REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	14.Sep.06	THESIS

4. TITLE AND SUBTITLE
A NEURAL RELEVANCE MODEL FOR FEATURE EXTRACTION FROM HYPERSPECTRAL IMAGES, AND ITS APPLICATION IN THE WAVELET DOMAIN.

5. FUNDING NUMBERS

6. AUTHOR(S)
MAJ MENDENHALL MICHAEL J

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RICE UNIVERSITY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI04-1877

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

14. SUBJECT TERMS

15. NUMBER OF PAGES
141

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

RICE UNIVERSITY

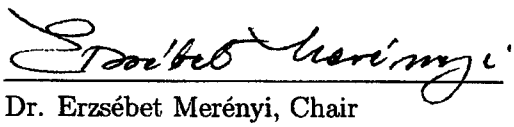
A Neural Relevance Model for Feature Extraction from Hyperspectral Images, and its Application in the Wavelet Domain

by

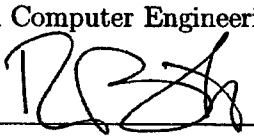
Michael J. Mendenhall

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree
Doctor of Philosophy

Approved, Thesis Committee:

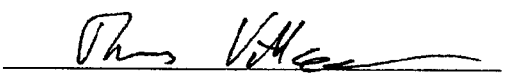


Dr. Erzsébet Merényi, Chair
Research Professor of Electrical
and Computer Engineering



Dr. Richard G. Baraniuk
Victor E. Cameron Professor of Electrical
and Computer Engineering

Dr. Devika Subramanian
Professor of Computer Science



Dr. rer. nat. habil. Thomas Villmann
Clinic for Psychotherapy, University Leipzig

Houston, Texas
August, 2006

Abstract

A Neural Relevance Model for Feature Extraction from Hyperspectral Images, and its Application in the Wavelet Domain

by

Michael J. Mendenhall

Our research is motivated by military applications related to aspects of contingency planning. Of recent interest is the identification of landmasses which can support the landing and takeoff of fixed wing and rotary aircraft where accurate classification of the surface cover is of utmost importance.

In a supervised classification scenario, a natural question is whether a subset of the input features (spectral bands) could be used without degrading classification accuracy. Our interest in feature extraction is twofold. First, we desire a significantly reduced set of features by which we can compress the signal. Second, we desire to enhance classification performance by alleviating superfluous signal content. Feature extraction models based on PCA or wavelets judge feature importance by the magnitude of the transform coefficients, rarely leading to an appropriate set of features for classification.

We analyze a recent neural paradigm, Generalized Relevance Learning Vector Quantization (GRLVQ) [1], to discover input dimensions relevant for classification. GRLVQ is based on, and substantially extends, Learning Vector Quantization (LVQ) [2] by learning relevant input dimensions while incorporating classification accuracy in the cost function. LVQ is the supervised version of Kohonen's unsupervised Self-Organizing Map [2]. LVQs iteratively

adjust prototype vectors to define class boundaries while minimizing the Bayes risk. Our analysis reveals two major algorithmic deficiencies of GRLVQ. Fixing these deficiencies leads to improved convergence performance and classification accuracy. We call our improved version GRLVQ-Improved (GRLVQI). By using only the relevant spectral channels discovered by GRLVQ, we show that one can produce as good or better classification accuracy as by using all spectral channels. We support this claim by running an independent classifier on the reduced feature set, using 23 classes of a real 194-band remotely sensed hyperspectral image. The higher the data dimension and/or larger the number of classes, the more advantage GRLVQI shows over GRLVQ.

The improved performance of GRLVQI over GRLVQ is substantiated using several different methods discussed in the literature. We come to the important conclusion that the improved results obtained by our GRLVQI are statistically significant.

A new and exciting feature extraction model is presented by applying GRLVQI in the wavelet domain. Our model is focused on classification requirements, rather than signal reconstruction. It does not follow the largest magnitude coefficient selection as is more typical in wavelet analysis. The most relevant wavelet features turn out to be something different. Further, it allows for a linearly selection of wavelet coefficients based on their computed relevances. We extend this work to complex wavelets in order to mitigate the effects of discontinuities introduced in the spectra due to the deletion of spectral bands containing irrecoverably corrupted data. The Dual-Tree Complex Wavelet Transform shows improved classification results with similar feature extraction capabilities as with the Critically Sampled Discrete Wavelet Transform. Our results demonstrate the superior classification and feature reduction performance of our relevance-wavelet model.

Abstract

A Neural Relevance Model for Feature Extraction from Hyperspectral Images, and its Application in the Wavelet Domain

by

Michael J. Mendenhall

Our research is motivated by military applications related to aspects of contingency planning. Of recent interest is the identification of landmasses which can support the landing and takeoff of fixed wing and rotary aircraft where accurate classification of the surface cover is of utmost importance.

In a supervised classification scenario, a natural question is whether a subset of the input features (spectral bands) could be used without degrading classification accuracy. Our interest in feature extraction is twofold. First, we desire a significantly reduced set of features by which we can compress the signal. Second, we desire to enhance classification performance by alleviating superfluous signal content. Feature extraction models based on PCA or wavelets judge feature importance by the magnitude of the transform coefficients, rarely leading to an appropriate set of features for classification.

We analyze a recent neural paradigm, Generalized Relevance Learning Vector Quantization (GRLVQ) [1], to discover input dimensions relevant for classification. GRLVQ is based on, and substantially extends, Learning Vector Quantization (LVQ) [2] by learning relevant input dimensions while incorporating classification accuracy in the cost function. LVQ is the supervised version of Kohonen's unsupervised Self-Organizing Map [2]. LVQs iteratively

adjust prototype vectors to define class boundaries while minimizing the Bayes risk. Our analysis reveals two major algorithmic deficiencies of GRLVQ. Fixing these deficiencies leads to improved convergence performance and classification accuracy. We call our improved version GRLVQ-Improved (GRLVQI). By using only the relevant spectral channels discovered by GRLVQ, we show that one can produce as good or better classification accuracy as by using all spectral channels. We support this claim by running an independent classifier on the reduced feature set, using 23 classes of a real 194-band remotely sensed hyperspectral image. The higher the data dimension and/or larger the number of classes, the more advantage GRLVQI shows over GRLVQ.

The improved performance of GRLVQI over GRLVQ is substantiated using several different methods discussed in the literature. We come to the important conclusion that the improved results obtained by our GRLVQI are statistically significant.

A new and exciting feature extraction model is presented by applying GRLVQI in the wavelet domain. Our model is focused on classification requirements, rather than signal reconstruction. It does not follow the largest magnitude coefficient selection as is more typical in wavelet analysis. The most relevant wavelet features turn out to be something different. Further, it allows for a linearly selection of wavelet coefficients based on their computed relevances. We extend this work to complex wavelets in order to mitigate the effects of discontinuities introduced in the spectra due to the deletion of spectral bands containing irrecoverably corrupted data. The Dual-Tree Complex Wavelet Transform shows improved classification results with similar feature extraction capabilities as with the Critically Sampled Discrete Wavelet Transform. Our results demonstrate the superior classification and feature reduction performance of our relevance-wavelet model.

Abstract

A Neural Relevance Model for Feature Extraction from Hyperspectral Images, and its Application in the Wavelet Domain

by

Michael J. Mendenhall

Our research is motivated by military applications related to aspects of contingency planning. Of recent interest is the identification of landmasses which can support the landing and takeoff of fixed wing and rotary aircraft where accurate classification of the surface cover is of utmost importance.

In a supervised classification scenario, a natural question is whether a subset of the input features (spectral bands) could be used without degrading classification accuracy. Our interest in feature extraction is twofold. First, we desire a significantly reduced set of features by which we can compress the signal. Second, we desire to enhance classification performance by alleviating superfluous signal content. Feature extraction models based on PCA or wavelets judge feature importance by the magnitude of the transform coefficients, rarely leading to an appropriate set of features for classification.

We analyze a recent neural paradigm, Generalized Relevance Learning Vector Quantization (GRLVQ) [1], to discover input dimensions relevant for classification. GRLVQ is based on, and substantially extends, Learning Vector Quantization (LVQ) [2] by learning relevant input dimensions while incorporating classification accuracy in the cost function. LVQ is the supervised version of Kohonen's unsupervised Self-Organizing Map [2]. LVQs iteratively

adjust prototype vectors to define class boundaries while minimizing the Bayes risk. Our analysis reveals two major algorithmic deficiencies of GRLVQ. Fixing these deficiencies leads to improved convergence performance and classification accuracy. We call our improved version GRLVQ-Improved (GRLVQI). By using only the relevant spectral channels discovered by GRLVQ, we show that one can produce as good or better classification accuracy as by using all spectral channels. We support this claim by running an independent classifier on the reduced feature set, using 23 classes of a real 194-band remotely sensed hyperspectral image. The higher the data dimension and/or larger the number of classes, the more advantage GRLVQI shows over GRLVQ.

The improved performance of GRLVQI over GRLVQ is substantiated using several different methods discussed in the literature. We come to the important conclusion that the improved results obtained by our GRLVQI are statistically significant.

A new and exciting feature extraction model is presented by applying GRLVQI in the wavelet domain. Our model is focused on classification requirements, rather than signal reconstruction. It does not follow the largest magnitude coefficient selection as is more typical in wavelet analysis. The most relevant wavelet features turn out to be something different. Further, it allows for a linearly selection of wavelet coefficients based on their computed relevances. We extend this work to complex wavelets in order to mitigate the effects of discontinuities introduced in the spectra due to the deletion of spectral bands containing irrecoverably corrupted data. The Dual-Tree Complex Wavelet Transform shows improved classification results with similar feature extraction capabilities as with the Critically Sampled Discrete Wavelet Transform. Our results demonstrate the superior classification and feature reduction performance of our relevance-wavelet model.

Disclaimer: The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

Acknowledgments

Personal drive, desire, and vision is what it takes to be successful. However, without help along the way to keep you motivated when things are looking down, to spark your interest, and to help you focus your vision, success is much more difficult. Having the opportunity to thank those that have helped along the way is the best part of the thesis.

I would like to thank my parents for taking an interest in what I do (and what I've done, good or bad) and making sure I had every opportunity early on to succeed in life.

Many thanks to my best friend and wife, who has been there through thick and thin. Without her help and support the Ph.D. process would have been an insurmountable task.

Thanks to my M.S. advisor, Roger, for setting me on this track and all his help getting the ball rolling. I am also indebted to my friend and former boss, Dallas, for helping me pursue this career path. Without his support, who knows where I would have ended up!

To Paul, my closest of friends, you have been an important part of my post secondary education experience from day one. Thanks for all of your good advice early on in the PhD process, it has been a big help. You have been a cherished friend for the better part of a decade, and I look forward to our many adventures to come!

Thanks to Ray at AFRL/SNJT for his support and funding. You have provided me with many wonderful opportunities to gain experience in remote-sensing and attend educational conferences for which I am grateful.

Thanks to Dr. Rich Baraniuk for playing an integral role in my successful completion. It is really quite an honor having you on my committee. Roger was a darn good advisor, I must have you to thank for that!

Dr. Thomas Villmann is an old colleague of Erzsébets, and a new one to me. Thanks for your guidance and input very early stages of searching for a research focus. I hope to have

the opportunity to work with you in the future.

To Dr. Devika Subramanian, many thanks for your helpful comments and wonderful input during the proposal. Thank you also for serving on my committee, it has been a joy.

Thanks to Mark Landers, CEO of SpecTIR. You have been terrific, and I appreciate the internship opportunities with Dr. Amer Smailbegovic. Amer has been a good friend over the last 3 years and has been a mentor to me in the remote sensing field.

Thanks to the many wonderful students at Rice University: Chris Stegar, Ryan King, Farbod, Ray Wagner, Ilan Goodman, Mona Sheik, Kadim Tasdemir, Lili Zhang, Chris Rozell, Mike Wakin, and Mike Lexa, to name a few.

Finally, I wish to take the opportunity to acknowledge and thank Dr. Erzsébet Merényi. I have had the opportunity to work for many folks over the past decade; few can compete. Your mentorship, your direction, your patience, and your friendship have made my Ph.D. experience a fun and exciting three years. I hope we may continue working together in the future.

Contents

Illustrations

Tables

Chapter 1

Introduction

1.1 Military requirements for imaging systems

Military intelligence and planning relies on aerial and satellite reconnaissance data for a multitude of planning and reporting functions including: target analysis and reporting, battle damage assessment, identification of enemy threats, and forward troop deployment planning. More recently, these communities are interested in assessing surface cover composition and terrain flatness to locate regions capable of supporting the landing and takeoff of fixed wing and rotary aircraft for two scenarios. The first scenario requires real-time assessment for emergency landing situations, for rapid deployment of special forces and their equipment to high-threat areas, and for "quick-looks" of the terrain to guide reconnaissance efforts. The second scenario is long term planning functions where one is interested in mapping potential landing sites for future troop deployments. We developed the Airfield Confidence Map (ACM) concept that proposes a solution for this second scenario [3].

1.2 Why traditional imagery will not do – hyperspectral imagery is a must

Many of the tasks discussed above are currently accomplished using imagery with high spatial resolution often necessitating low altitude aerial reconnaissance missions. However, such imagery is often unattainable due to enemy threats. Planning tasks can be greatly enhanced with remotely sensed hyperspectral data as high spectral resolution imagery (with potentially hundreds of spectral bands) provide all discriminating details needed for fine delineation of many material classes offering flexibility where purely spatial data falls short. We describe briefly how spectral data can overcome the pitfalls of purely spatial data as it

pertains to the tasks defined above.

- The detection of small targets is nearly impossible without high spatial resolution imagery. Due to the dangers of low altitude reconnaissance flights, we are forced to use high altitude imagery with low spatial resolution. In this scenario, high spectral resolution data *can* be used to detect sub-pixel sized targets if the spectral resolution is high. For the safety of troops and equipment, target assessment necessitates high spectral resolution data for target analysis and reporting.

- Spectral information can be used to detect and identify dangerous airborne chemicals before sending in ground troops to perform a detailed battle damage assessment. Spatial data cannot be used for such a function.

- The assessment of small guerrilla cells threatening operations can be detected by spectral signatures unique to enemy clothing, camouflage paints, and equipment where they are likely invisible to a spatial-only sensor.

- Troop deployments can be aided by identifying trafficable regions thus ensuring troops and equipment have the safest route from their staging point to their forward deployed location.

- Finally, the identification of surface cover is a key component of the Airfield Confidence Map problem, a task for which spatial data alone cannot provide a solution.

We focus on hyperspectral imagery for the many reasons summarized above.

The use of unmanned aerial vehicles (UAV) is ever increasing for reconnaissance efforts because it is cost effective and more importantly, there is no danger of human loss. We envision next-generation UAVs with high spectral resolution sensors for reconnaissance purposes. Fig. 1.1 depicts the acquisition of a hyperspectral scene by a UAV, which then transmits processed data to an Air Operations Center (AOC.) Using the newly acquired data, AOC personnel are ready to carry out the intelligence and planning functions we described earlier.

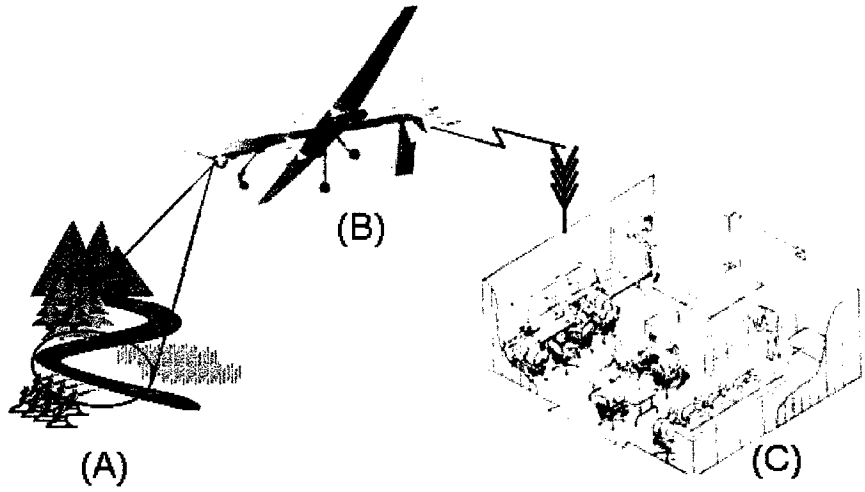


Figure 1.1 : Surface information **(A)** is gathered via unmanned aerial vehicle (UAV) **(B)** and sent to the air operations center (AOC) **(C)** for analysis in real time. The predator UAV image **(B)** was obtained from http://www.aeronaut.ru. Image **(C)** is a conceptual drawing of the AN/TSQ-165 modular AOC (MAOC) for deployed operations and was obtained from http://www.fas.org.

1.3 Classifier requirements – flexibility, accuracy, and feature extraction

We presented two scenarios under which we can operate with the ACM; a concept we developed in an earlier work [3] with follow-on field analysis [4]. In the first scenario, we are interested in the real-time assessment of surface cover for emergency landing and the rapid deployment of troops, and for "quick-looks" to guide reconnaissance efforts. For this function, we need a classifier on board the UAV so we can transmit the class labels. In non-emergency cases, if we are only interested in a few classes, and we have poor communications conditions, it might be more feasible to classify on-board the UAV and send the class labels to the AOC. If we are interested in a large number of classes and we have adequate communications conditions, it could require less processing time to send a reduced set of features (if our feature extraction is fast), allowing us to classify the data as it is being received at the AOC. Whatever the conditions, we need a flexible classifier we can adapt to different real-time situations without adversely affecting classification accuracy. At the

same time, we may want to continuously train our classifier with newly acquired samples to capture a potentially changing environment. If intelligence data indicates the need to look for additional surface materials, we need a classifier with the flexibility to add one or more classes on-the-fly without having to discard the state of the current classifier.

The requirements for long-term planning are different. This function is executed upon the completion of a reconnaissance mission and after data is retrieved from the UAV. In a deployment planning scenario, one would be interested in charting a large number of potential landing zones requiring perhaps thousands of hyperspectral scenes. The sheer volume of hyperspectral data would make classification an extremely long process. One common solution is to add computing resources in order to reduce the processing time. However, in a deployed environment, computing resources are often scarce. For processing efficiency, we desire a reduced feature set to classify the hyperspectral scenes. We are not, however, willing to use a reduced feature set at the expense of degraded classification accuracy.

Our discussion leads us to the following critical question:

> *Can we define a unified feature extraction and classification model for hyperspectral data with the constraint that we can achieve the same (or better) classification accuracy for a specific set of material classes of interest, as can be achieved with the original unaltered data?*

1.4 Thesis organization

This thesis effort will answer the stated research question with a *resounding yes*! The thesis document consists of seven chapters. Each chapter is arranged in such a manner as to allow those with current knowledge of certain theoretical aspects to easily skim those sections and quickly focus on the important contributions of this work.

Chapter 1 motivates the use of hyperspectral images as an aide to military planners.

Our work is not specific to one small subset of applications but rather has a broad audience. We discussed several critical tasks for which regular photographic-type imagery is ill-suited. The presentation of several specific tasks leads us to a set of very broad set of classifier requirements. Finally, we propose a challenging research question; the remainder of this thesis is focused on answering that question.

Chapter 2 presents two philosophies of feature extraction and classification processes. We provide background information on the feature extraction and classification paradigm of Generalized Relevance Learning Vector Quantization (GRLVQ). An original analysis of the update window of GRLVQ and its predecessor (LVQ2.1) are discussed in light of a common misconceptions regarding the importance of these windows. In our investigation, we find that GRLVQ suffers from two deficiencies which we address in our GRLVQ-Improved (GRLVQI).

Chapter 3 provides a more detailed discussion of hyperspectral images and some interesting applications to illustrate the flexibility of this coveted data. The Lunar Crater Volcanic Field (LCVF) data set is discussed in detail as is the design of the classification experiments. Specific design considerations for an LVQ network are presented along with a review of current literature to gain insights on design guidance. Finally, we review a challenging benchmark classification of the LCVF data by a hybrid artificial neural network so we may evaluate the classification capabilities of the baseline GRLVQ and our improved GRLVQI.

Chapter 4 presents the results of our experiments. We start with background information on how we measure classification success and how we determine the number of important features as well as methodology on an independent evaluation of the discrimination capability of those features. Finally, we present the GRLVQ(I) results on the LCVF data set in two parts. Part I is an evaluation of classifier performance while Part II evaluates the feature extraction capabilities. A manuscript submitted for publication [5] is comprised of our original analysis, contributions to GRLVQ (GRLVQI), and results (material from Chapters 2 - 4).

Chapter 5 answers the following question: "Are the improvements of GRLVQI over GRLVQ significant?" We delve into a comparison of the classification performance of GRLVQI versus GRLVQ. We analyze the significance of the improvements using several techniques presented in the literature. We come to the important conclusion that the classification improvements exhibited by GRLVQI over GRLVQ on hyperspectral data are indeed significant. A majority of this chapter's material (less discussion on generalization bounds) was recently submitted for publication [6].

Chapter 6 is a rather large chapter which contributes greatly to the feature reduction aspect of our research question. This chapter consists of two major parts. Part I investigates the Critically Sampled Discrete Wavelet Transform (CSDWT) for its sparseness and nearly decorrelated coefficients as a feature space for GRLVQI processing. Results of GRLVQI processing on the CSDWT representation of the data are presented and potential issues with data discontinuities are discussed. The information covered to this point were presented at a remote sensing conference and published in its proceedings [7]. Part II provides an in-depth look at the problem of data discontinuities and their effect on the wavelet coefficients. We investigate the magnitude of the Dual-Tree Complex Wavelet Transform as a feature space for GRLVQI processing. We find that the odd basis functions of the imaginary component leads to better classification accuracy with similar feature reduction performance as the CSDWT. Part II of this chapter was presented at the *IEEE Mountain Workshop on Adaptive and Learning Systems* conference and published in their proceedings [8].

Finally, in Chapter 7 we summarize our research and our important contributions. Although we discuss potential areas of continued research at the end of each chapter, we also provide a brief review of topics in this summary chapter. Finally, we present areas of potential research that were not covered in the main body of the thesis document.

Chapter 2

Joint classification and feature extraction

This chapter motivates the use of joint feature extraction and classification methods and in particular, neural learning paradigms. We briefly discuss works related to our research focus and conclude that Generalized Relevance Learning Vector Quantization (GRLVQ), an adaptive neural learning paradigm, realizes a joint classification and feature extraction paradigm which is appropriate for processing very high-dimensional data. We present the GRLVQ algorithm and formulae, then provide an original analysis of the windowing effects of Learning Vector Quantization (LVQ)2.1 (GRLVQ's predecessor) relating discussion in the literature on the "effective" update window for GRLVQ based on prototype vector update rules. Keeping with the traditional view that the window in LVQ2.1 is used to promote decision boundary development, we consider further restricting the update window of GRLVQ, to do the same. This leads us to an important conclusion that restricting the prototype vector updates by a window is not necessary, and can prohibit the development of the classifier. Our original analysis of GRLVQ reveals two critical issues: potential for diverging prototype vectors and poor prototype utilization. We address these two issues as our contribution to GRLVQ yielding our improved version, GRVQ-Improved (GRLVQI).

2.1 Motivation and Background

Remotely sensed hyperspectral images are sophisticated data sets that provide the discrimination capability needed for the military planning functions discussed in Chapter 1. The intricacy of this data pose very challenging problems for pattern classification algorithms, classification being its primary use. For air planners to extract the full potential from hy-

perspectral images coupled with the challenges in accurate material identification, signal processing techniques with the utmost sophistication are required. We seek a signal processing "edge" so we can more easily and more accurately distinguish the material classes of interest. One way to achieve such an edge is to reduce the information to exactly that which is needed for classification. That is, feature extraction can play a key role in processing hyperspectral signatures by reducing the superfluous signal content while preserving that part of the signal important for classification.

Given a particular image where several classification problems may be drawn (e.g., a study on vegetation for a specific image would likely yield a different set of classes than a soil study for that same image), the optimal set of features for classification will be different for each study, regardless of whether or not the image is the same. That is to say, it is not reasonable to assume that a global set of features exists for all hyperspectral images with any arbitrary number of classification problems. An example of which would be using as features the N largest magnitude wavelet coefficients (or wavelet subband energy). However, several studies indicate that the N largest magnitude wavelet coefficients are not an adequate feature set for classification [7, 9, 10]. One should be aware that certain feature extraction methods well suited for particular applications are often ineffective for preserving information for classification (e.g., largest magnitude wavelet selection is well suited to signal reconstruction and denoising, yet ineffective for selecting features for classification).

Several feature extraction and classification processes are available. We categorize them as independent processes (i.e., where feature extraction is accomplished independently of classification as in [9–11]) or joint processes (where feature extraction and pattern classification are intertwined process as in [1, 5, 12]). The former can lead to inferior results where classification is likely hindered because the wrong information is retained. This deteriorated performance occurs because the feature extraction process is unaware of what is important

for the classifier to distinguish class structure. The latter, in our opinion, offers the greatest chance at successful classification and feature extraction because there is an interplay between the two processes.

Many current methods claiming joint or unified feature extraction and classification do not optimize the former for the latter [12]. According to Oehler and Gray [12], some optimize for compression of several different signals (e.g., in speech processing) where the compressor yielding the smallest distortion indicates the class to which the signal belongs. Others optimize for compression first, then cascade the compressed output to a classifier that minimizes the probability of error based on the feature set. Oehler and Gray present a true joint compression and classification system which uses a Learning Vector Quantization (LVQ) algorithm to minimize a distortion function which includes a *squared error* term and a *Bayes risk* term in a Lagrange multiplier form [12]. Although this specific method fits well with our requirements, a parametric model for the posterior class probabilities is needed. Without the posterior class probabilities, the distortion function degrades to a squared error term.

As with any task, one must select the right tool for the job. In order to do this, one must understand the specific challenges of the problem at hand. Remotely sensed hyperspectral data is particularly difficult to process and makes appropriate selection of the joint feature extractor and classifier hard. It is most often the case that parametric models do not exist. As such, we are unable to use methods that require parametric models for class probabilities (e.g., [12]). Measurement noise and missing or incomplete data also influence which joint feature extraction and classification algorithms may be of use. Further, it is common to have relatively few training samples, often fewer than the number of dimensions in the spectra. The presence of rarely occurring material types within a particular scene further complicates matters. It is most often the case where each class does not have at least one more training sample than input dimensions so methods such as Maximum Likelihood classification are

completely removed from our signal processing tool box. Those methods that require the entire training sample pool to have at least one more training sample than the number of input dimensions (such as Principal Components) often do not equally represent all classes and do not work well for hyperspectral data (see e.g., [13] and Section 6.6.)

In our scenario, we make the following assumptions. First, a known mapping between the acquisition system of the newly acquired reconnaissance data and the acquisition system of the training data exists. Training data are often obtained from the same sensor used during acquisition making this first assumption automatically valid. Second, atmospheric conditions of both acquisition and training data are the same or at least known, and therefore can be corrected for. Records of weather conditions across the globe exist, so it is reasonable to assume under which conditions the different data are acquired. Third, we have a (potentially large) set of predefined classes (surface materials) of interest. Planning for a specific task will dictate the use of the imagery and necessitate a defined set of classes validating this final assumption.

Based on our requirements from Chapter 1 and our assumptions stated above, we consider the doubly adaptive neural learning paradigm of Generalized Relevance Learning Vector Quantization (GRLVQ) [1], by Hammer and Villmann, for classification-driven feature extraction. The GRLVQ adapts a set of prototype vectors to define classification boundaries while adapting a weighting of the input dimensions to reflect what was important in classification. In our analysis of GRLVQ, we discover two deficiencies: instability and poor resource allocation for large, complex data sets such as hyperspectral data. We remedy those deficiencies resulting in faster convergence and better classification accuracy.

2.2 Learning Vector Quantization (LVQ) for feature extraction

Kohonen's Learning Vector Quantization 2.1 (LVQ2.1) [2] is a supervised neural learning paradigm where each class is assigned a fixed number of prototype vectors. These prototype vectors are iteratively adjusted to represent (model) the classes. LVQ2.1 variants position prototype vectors by *differentially shifting* (Fig. 2.1) a best matching (winning) prototype vector with the same label as the input training sample and a best matching (winning) prototype vector with a different label than the input training sample, at each iteration.

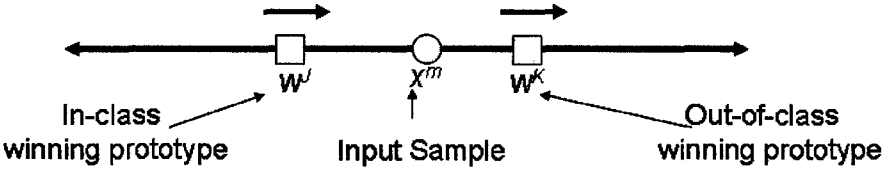


Figure 2.1 : Differential shifting of the in-class winning prototype vector ($\mathbf{w}^J$) and the out-of-class winning prototype vector ($\mathbf{w}^K$).

After a sufficient number of adaptation steps, prototype vectors are placed to define classification boundaries while minimizing the Bayes risk. LVQs belong to a class of maximal-margin algorithms that maximize the hypothesis margin [14] and certain forms of LVQ (Generalized LVQ and Generalized Relevance LVQ for example) are *gradient descent* algorithms with the following general update form for the weights (prototype vectors) [15]:

$$w(t+1) \;=\; w(t) - \epsilon(t)\nabla C(w(t)), \tag{2.1}$$

where $C(w(t))$ is the cost function we wish to minimize, $\epsilon(t)$ is the learn parameter (or learn rate), $w(t)$ is the state of the prototype vector at time t, and ∇ is the gradient operator.

Sato and Yamada [16] developed the Generalized LVQ (GLVQ) to address divergence issues issues exhibited by LVQ2.1. The specific issue is one of prototypes drifting from their optimal locations once placed which results in a degradation in classification accuracy. This divergence issue is addressed by descending a cost function C which includes a measure of the misclassifications.

Using GLVQ as the starting point, Hammer and Villmann [1] incorporate an importance weighting of the input dimensions for classification. This adaptive diagonal metric is the so-called relevance and gives us Generalized Relevance LVQ (GRLVQ). The GRLVQ is then a double adaptive neural learning paradigm. Prototype vectors are adapted, in an iterative fashion, to define classification boundaries. While prototypes learn class structure, the relevance adapts to reflect which input dimensions are most important for the given classification.

Let us facilitate our discussion of GRLVQ by defining variables similar to [1]:

- Define the training sample set as $\{\mathbf{x}^m, y^m\}_{m=1}^{M} \in \{\Re^n \times \Re\}$. There are M samples x with n dimensions and class labels y.
- Define $\{W\}$ as the set of all prototype vectors where $\mathbf{w}^J \in \{W\}$ is the best matching in-class prototype vector with class label y^m, the same as that of the input sample $\mathbf{x}^m$. The number of prototypes for class y^m is P and prototypes in class y^m are indexed by $p = \{1, \ldots, P\}$. Further define $\mathbf{w}^K \in \{W\}$ as the best matching out-of-class prototype vector with class label $y^r \neq y^m$.
- Define d^J and d^K as the squared Euclidean distance between the input sample $\mathbf{x}^m$ and prototype vectors $\mathbf{w}^J$ and $\mathbf{w}^K$, respectively. The notation d^p is the squared Euclidean distance between prototype vector $\mathbf{w}^p$, $p \in \{1, ..., P\}$ with class label y^m, and the sample $\mathbf{x}^m$.
- Define $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ as an n-dimensional vector of relevance factors and Λ as a diagonal matrix with $\Lambda_{ii} = \lambda_i$ where $i \in \{1, ..., n\}$.
- Define the weighted squared Euclidean distance between the input sample $\mathbf{x}^m$ and prototype $\mathbf{w}^J$ and $\mathbf{w}^K$ as d_λ^J and d_λ^K, respectively, where λ indicates relevance factors used in the Euclidean distance calculation.
- Denote $\mu(\mathbf{x}^m)$ as the misclassification measure.
- Denote $f(\mu(\mathbf{x}^m))$ as the loss function.
- Denote C as the cost function.

- Define $\mathbf{M}^{JK} = \frac{\mathbf{w}^J + \mathbf{w}^K}{2}$, the midpoint between $\mathbf{w}^J$ and $\mathbf{w}^K$. Further define d^{JK} as the squared Euclidean distance between $\mathbf{w}^J$ and $\mathbf{w}^K$.

Definitions for the misclassification measure, loss function, and cost function are given in Eq. 2.2, Eq. 2.3, and Eq. 2.5 respectively.

There are two aspects to the gradient descent problem of Eq. 2.1. The first is to define a *misclassification measure* to represent a correct versus wrong classification. Sato and Yamada define the misclassification measure as:

$$\mu(\mathbf{x}^m) = \left(\frac{d^J - d^K}{d^J + d^K}\right). \tag{2.2}$$

In Eq. 2.2, $\mu(\mathbf{x}^m)$ is a normalized distance bounded by -1 and 1. This definition has a nice numerical interpretation of how well the current sample was classified. A correct classification occurs if $\mu(\mathbf{x}^m) < 0$ and the sample is classified perfectly if $\mu(\mathbf{x}^m) = -1$. A wrong decision is made if $\mu(\mathbf{x}^m) \geq 0$.

The second aspect is to define a differentiable *loss function* that takes into account the misclassification measure $\mu(\mathbf{x})$. Sato and Yamada define the loss function as:

$$\begin{aligned} f(\mu(\mathbf{x}^m)) &= \frac{1}{1 + e^{-\mu(\mathbf{x}^m)}}, \\ &= \frac{1}{1 + e^{\left(-\frac{d^J - d^K}{d^J + d^K}\right)}}. \end{aligned} \tag{2.3}$$

Using the sigmoid function (Fig. 2.2 top) as the loss function (Eq. 2.3) has the distinct advantage of having a derivative (Fig. 2.2 bottom) that is a function of itself:

$$\begin{aligned} f'(\mu(\mathbf{x}^m)) &= f(\mu(\mathbf{x}^m))\left[1 - f(\mu(\mathbf{x}^m))\right], \\ &= \frac{e^{\left(-\frac{d^J - d^K}{d^J + d^K}\right)}}{\left(1 + e^{\left(-\frac{d^J - d^K}{d^J + d^K}\right)}\right)^2}. \end{aligned} \tag{2.4}$$

The *cost function* C minimized in Eq. 2.1 is a simple sum of the loss for each sample [16]:

$$C = \sum_{m=1}^{M} f(\mu(\mathbf{x}^m)). \tag{2.5}$$

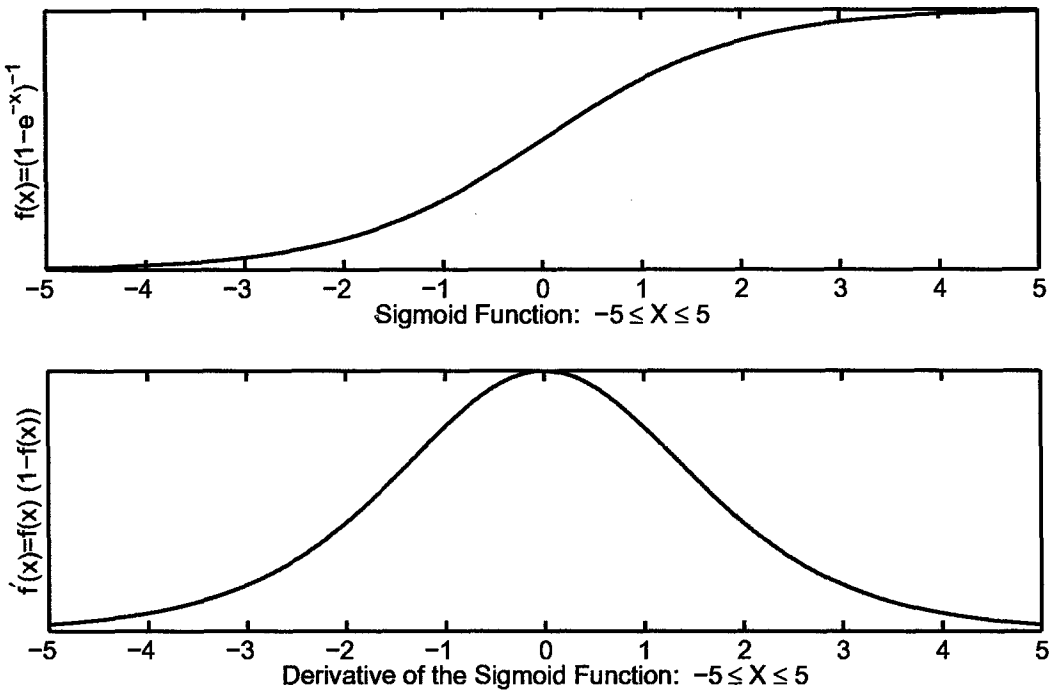


Figure 2.2 : **Top:** The sigmoid function. **Bottom:** Derivative of the sigmoid function.

Hammer and Villmann [1] use as the basis for their algorithm, the misclassification measure, loss function, and cost function described above. One can change the behavior of the algorithm by defining a different loss function or a different misclassification measure (see, e.g., Sato and Yamada [17] and Juang and Katagiri [18]).

2.2.1 Generalized Relevance LVQ (GRLVQ) winner selection

The index of the winning prototype vector, $\mathbf{w}^c$, is selected as

$$c = \arg\min_q \left(\sum_{i=1}^{n} \lambda_i (x_i^m - w_i^q)^2 \right). \tag{2.6}$$

Winner selection in GRLVQ is performed twice, once where $\mathbf{w}^q$ has class label $y^q = y^m$, and a second time where $\mathbf{w}^q$ has class label $y^q \neq y^m$, to give both an in-class winning prototype vector ($\mathbf{w}^J$) *and* out-of-class winning prototype vector ($\mathbf{w}^K$) respectively. The squared Euclidean distances, weighted using the relevance factors λ_i, between the input

sample $\mathbf{x}^m$ and prototypes $\mathbf{w}^J$ and $\mathbf{w}^K$ are:

$$d_\lambda^J = \sum_{i=1}^{n} \lambda_i (x_i^m - w_i^J)^2, \tag{2.7}$$

$$d_\lambda^K = \sum_{i=1}^{n} \lambda_i (x_i^m - w_i^K)^2. \tag{2.8}$$

2.2.2 GRLVQ prototype updates – descending a cost function

Prototype vectors learn in an iterative fashion to define boundaries between neighboring classes using the differential shifting prototype update strategy of LVQ2.1 [2]. The best matching in-class prototype vector ($\mathbf{w}^J$) is moved toward the input training sample and the best matching out-of-class prototype vector ($\mathbf{w}^K$) is moved away from the input training sample, regardless of a correct or incorrect decision about class membership. Following the gradient descent form of Eq. 2.1, updates for the in-class and out-of-class winning prototype vectors in [1] are:

$$\Delta \mathbf{w}^J = \frac{4\epsilon(t)^J f'|_{\mu(\mathbf{x}^m)} d_\lambda^K}{(d_\lambda^J + d_\lambda^K)^2} \Lambda(\mathbf{x}^m - \mathbf{w}^J), \text{ and} \tag{2.9}$$

$$\Delta \mathbf{w}^K = -\frac{4\epsilon(t)^K f'|_{\mu(\mathbf{x}^m)} d_\lambda^J}{(d_\lambda^J + d_\lambda^K)^2} \Lambda(\mathbf{x}^m - \mathbf{w}^K), \tag{2.10}$$

where the loss function f is the sigmoid function (Eq. 2.3), f' the derivative of the sigmoid function (Eq. 2.4), $\epsilon(t)^J$ the in-class learn rate, and $\epsilon(t)^K$ the out-of-class learn rate.

2.2.3 Relevance updates – identifying important input features

Relevance factors hold the potential for dimensionality reduction by learning which input dimensions are important for classification. Importance is learned in a weight vector with values (computed by GRLVQ) called relevances (λ). The interpretation of the relevance is rather straightforward, input dimensions with larger relevances are interpreted as being more

important for classification than input dimensions with smaller relevances. We can select the most important input features to achieve a good classification by ordering features (input dimensions) based on their relevance.

The relevance factor updates are found via gradient descent in [1], similar to the prototype vector updates, giving the following update rule:

$$\lambda_i = \max\{\lambda_i - \frac{2\epsilon(t)^{\lambda} f'|_{\mu(\mathbf{x}^m)} \left(d_\lambda^K(x_i^m - w_i^J) - d_\lambda^J(x_i^m - w_i^K)\right)}{(d_\lambda^J + d_\lambda^K)^2}, 0\}. \tag{2.11}$$

Relevance factors are scaled such that $\|\lambda\|_1 = 1$ in order to avoid numerical instabilities [1]. Any p-norm may be used to scale λ, however, we choose the l_1-norm because it may have a convenient interpretation as a probability.

2.3 Original analysis of the LVQ2.1 and G(R)LVQ update windows

In this section we present and original analysis of the LVQ2.1 and G(R)LVQ update windows. We investigate the effects of further constraining the updates to $\mathbf{w}^J$ and $\mathbf{w}^K$ to reflect windowing philosophy discussed in the literature. Finally, we provide details on two problems with GRLVQ that lead the potential for prototype divergence and poor prototype utilization. Addressing these two issues is our contribution to the neural learning paradigm of GRLVQ.

2.3.1 Part I – windowing in LVQ2.1 and G(R)LVQ

The LVQ2.0 is the predecessor to LVQ2.1 where the difference between these two paradigms is the addition of an "update window". A window is used to influence the development of the decision boundary by (what is believed to be almost exclusively) focusing on input samples which lie in the middle of the decision boundary described by the current (fixed) state of the prototype vectors. This is consistent with a description of the LVQ2.1 update window as a step function about the mid-point of the in-class and out-of-class winning prototype

vectors [2, 17]. In this section, we look closer at the LVQ2.1 update window as originally defined by Kohonen and compare the window to that which is defined by the update rules in Eq. 2.9 and Eq. 2.10 for G(R)LVQ.

2.3.1.1 The LVQ2.1 update window

Recall that G(R)LVQ uses the differential shifting prototype update strategy of LVQ2.1 discussed in Section 2.2.2. In Kohonen's treatment of LVQ2.1, a window is used to promote development of the decision boundary. This window is often viewed as a step function (in the one-dimensional case) as portrayed in Fig. 2.3.

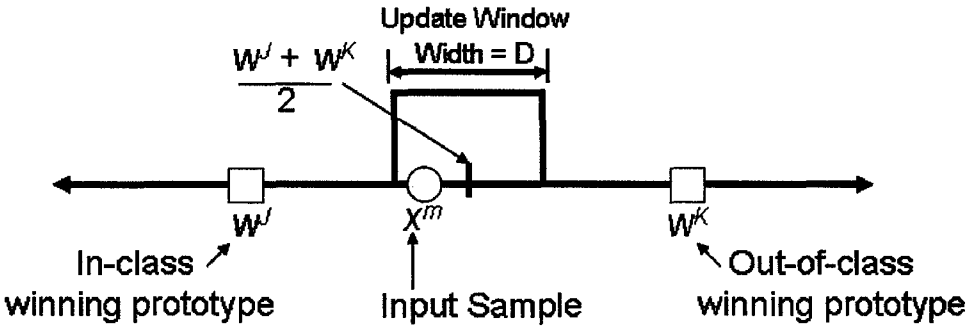


Figure 2.3 : The LVQ2.1 update window is often described as a step function centered between prototypes $\mathbf{w}^J$ and $\mathbf{w}^K$ with a window of width D, where D is the percentage of the distance between $\mathbf{w}^J$ and $\mathbf{w}^K$.

In Kohonen's definition of LVQ2.1, the in-class and out-of-class prototype vectors are updated if and only if the following condition holds [2]:

$$\min\left(\sqrt{\frac{d^J}{d^K}}, \sqrt{\frac{d^K}{d^J}}\right) > \frac{1-D}{1+D}. \tag{2.12}$$

The variable D in Eq. 2.12 represents a window of relative width which is the percentage of the Euclidean distance ($\sqrt{d^{JK}}$ between $\mathbf{w}^J$ and $\mathbf{w}^K$. For example, a window width $D = 0.5$ would occupy 50% of the this distance. We show that which is described by Eq. 2.12 is not a step function about $\mathbf{M}^{JK}$ (the midpoint of $\mathbf{w}^J$ and $\mathbf{w}^K$) as viewed by others [2, 17], it is something different.

We will show for a two-dimensional case that Eq. 2.12 defines a circular boundary around each prototype vector where input samples ($\mathbf{x}^m$) lying outside of the circle result in updates to both prototype vectors. Instead of focusing on what lies in or out of the circular boundary, we focus on the boundary itself. In a simple two-class problem with a single prototype vector assigned to each class, there are only two boundaries, one for the in-class prototype vector and one for the out-of-class prototype vector.

For the in-class prototype $\mathbf{w}^J$, we define the boundary by equating the left and right sides of Eq. 2.12, rewriting $\frac{1-D}{1+D}$ as $\frac{a}{b}$ where $a = (1 - D)$ and $b = (1 + D)$ for simplicity. If the sample is classified correctly, then $\frac{d^J}{d^K}$ (Eq. 2.12) is smallest and the boundary for $\mathbf{w}^J$ is:

$$\begin{aligned} \sqrt{\frac{d^J}{d^K}} &= \frac{1-D}{1+D}, \\ &= \frac{a}{b}. \end{aligned} \tag{2.13}$$

Squaring both sides of Eq. 2.13 yields:

$$\frac{d^J}{d^K} = \frac{a^2}{b^2}.$$

For our two-dimensional example, we can rewrite Eq. 2.14 as:

$$\frac{\left(w_1^J - x_1\right)^2 + \left(w_2^J - x_2\right)^2}{\left(w_1^K - x_1\right)^2 + \left(w_2^K - x_2\right)^2} = \frac{a^2}{b^2}.$$

After cross multiplying, collecting terms, and completing the square, it is readily apparent that the update boundary (about $\mathbf{w}^J$) is defined by a circle for the two-dimensional example presented here. The circular boundary is not centered at $\mathbf{w}^J$, rather it has its center located at

$$\left(-\frac{a^2 w_1^K - b^2 w_1^J}{b^2 - a^2}, -\frac{a^2 w_2^K - b^2 w_2^J}{b^2 - a^2}\right),$$

with a radius defined as

$$\left(\frac{a^2 w_1^K - b^2 w_1^J}{b^2 - a^2}\right)^2 + \left(\frac{a^2 w_2^K - b^2 w_2^J}{b^2 - a^2}\right)^2 - \frac{b^2\left(w_1^J + w_2^J\right) - a^2\left(w_1^K + w_2^K\right)}{b^2 - a^2}.$$

One can similarly define the circular boundary for the out-of-class prototype vector by equating $\sqrt{d^K/d^J}$ to $\frac{a}{b}$ and going through the same process as above.

For the two-class case, with a single prototype vector for each class, we demonstrated that Eq. 2.12 describes two circular regions, one about each prototype vector. Updates to the in-class and out-of-class winning prototype vectors occur if the input sample lies outside of the circular regions and *do not* occur if the sample falls inside the circular regions. Further, prototype vectors are not located at the centers of the boundaries, they are eccentrically located in equal distances but opposite directions (Fig. 2.4). The generalization of the two-class case defines hyperspherical regions about each prototype vector. For multiple classes with multiple prototype vectors per class, the results are similar but are further constrained by their multi-dimensional Voronoi cells.

We rewrite the LVQ2.1 prototype updates to include indicator functions that capture Kohonen's update rule. That is, $g_J(\cdot)$ and $g_K(\cdot)$ in Eq. 2.14 and Eq. 2.15 evaluate to 1 if the condition in Eq. 2.12 holds:

$$\Delta \mathbf{w}^J = \epsilon(t) g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)(\mathbf{x}^m - \mathbf{w}^J), \tag{2.14}$$

$$\Delta \mathbf{w}^K = -\epsilon(t) g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)(\mathbf{x}^m - \mathbf{w}^K). \tag{2.15}$$

In general, $g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)$ and $g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)$ are composed of all the information in update rules 2.9 - 2.10 except the learn rate and the difference between the prototype and the input sample. A two-dimensional example of the Kohonen window is presented in Fig. 2.4 for varying window widths. Black regions of Fig. 2.4 equate to $g_J(\cdot) = g_K(\cdot) = 0$ while white regions equate to $g_J(\cdot) = g_K(\cdot) = 1$. Prototypes are indicated as white dots in the black regions. Fig. 2.4 clearly shows that updates to $\mathbf{w}^J$ and $\mathbf{w}^K$ not only occur when the sample falls within a window about their midpoint, they are updated for a wide range of input samples. It is unclear what effect the Kohonen window has on the development of

the decision boundary for samples lying outside of the midpoint between $\mathbf{w}^J$ and $\mathbf{w}^K$. We surmise this contributes to the divergence problem LVQ2.1 exhibits.

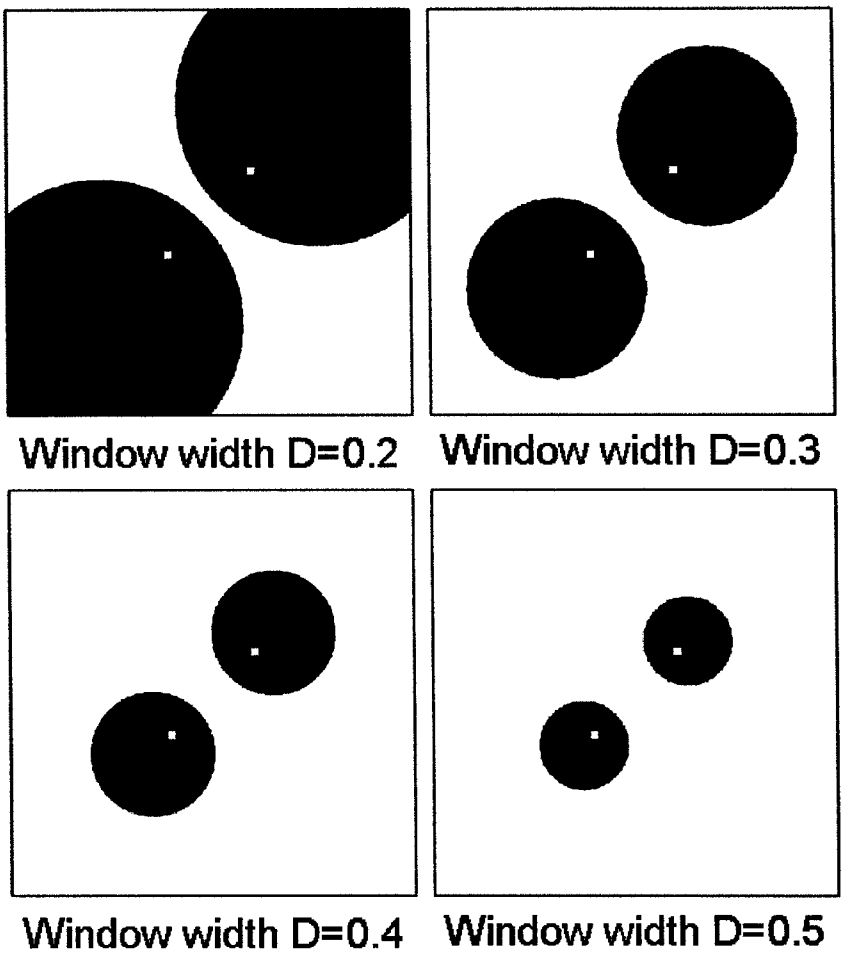


Figure 2.4 : Kohonen's window for two-dimensional data. For inputs in the black regions, prototype vectors *are not* updated; white regions are those areas where prototype vectors *are* updated. Prototypes are indicated by white dots located in the black regions.

2.3.1.2 The G(R)LVQ Gaussian-like update window

Sato and Yamada state that the update window in GLVQ is no longer needed because it is replaced by a Gaussian-like window [17]. From Eq. 2.4 and Fig. 2.2 bottom, it is clear the derivative of the sigmoid function is Gaussian-like. However, since the remaining pieces of the update are also a function of the input sample and locations of the prototype vectors,

it unclear at this time how the rest of the update formulation in Eq. 2.9 and Eq. 2.10 affect this Gaussian-like window found in G(R)LVQ.

In order to evaluate the effect of the G(R)LVQ update window, we define the two windowing functions $g_J(\cdot)$ and $g_K(\cdot)$ of Eq. 2.14 and 2.15 as:

$$g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K) = \frac{4d_\lambda^K f'|_{u(\mathbf{x}^m)}\Lambda}{(d_\lambda^J + d_\lambda^K)^2}, \tag{2.16}$$

$$g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K) = \frac{4d_\lambda^J f'|_{u(\mathbf{x}^m)}\Lambda}{(d_\lambda^J + d_\lambda^K)^2}. \tag{2.17}$$

To provide some illustration, we consider a 2-class problem in one dimension. The in-class prototype vector ($\mathbf{w}^J$) and the out-of-class prototype vector ($\mathbf{w}^K$) are fixed at locations -5 and 5 (indicated by dots in Fig. 2.5). The values of the windowing functions $g_J(\cdot)$ and $g_K(\cdot)$ are plotted on the y axis for different values of the input sample $\mathbf{x}^m$ along the x axis. Function values evaluated to the left of the line defined by the input sample $\mathbf{x}^m = \mathbf{0}$ are for a correct classification whereas the function values evaluated to the right are for a wrong decision. For the one-dimensional case, $\Lambda = [\lambda_{11}] = \mathbf{1}$.

For a correct decision, Fig. 2.5 shows that $\mathbf{w}^J$ is moved much closer to the sample than $\mathbf{w}^K$ is moved away. For the extreme case when $\mathbf{x}^m = \mathbf{w}^J$, no update occurs to $\mathbf{w}^K$. For a wrong decision, $\mathbf{w}^J$ is moved slightly closer to the sample whereas $\mathbf{w}^K$ is moved much further away. Prototype vectors are updated most when the sample falls in the areas slightly skewed from their midpoint. The graphical representation of the LVQ2.1 and G(R)LVQ windows (Fig. 2.4) and G(R)LVQ (Fig. 2.5), show that the latter can be fine-tuned to better promote the development of the decision boundary.

2.3.1.3 Restricting the G(R)LVQ update – emphasizing the midpoint between winning prototype vectors

We investigate several windowing strategies based on the assumption that a window aides in LVQ's decision boundary definition. The windows discussed in the following sections are not

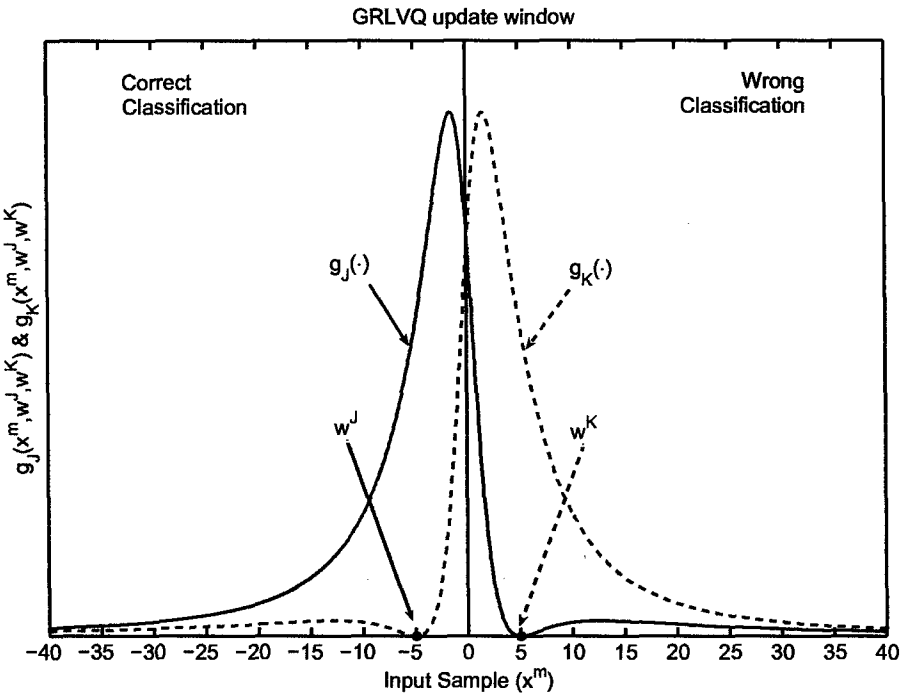


Figure 2.5 : Effective update window for GRLVQ for one-dimensional data. The values of $g_J(\cdot)$ and $g_K(\cdot)$ along the y axis are evaluated for fixed prototype locations at -5 ($\mathbf{w}^J$) and 5 ($\mathbf{w}^K$) where the input sample $\mathbf{x}^m$ varies along the x axis. Solid curves reflect updates to the in-class prototype vector ($\mathbf{w}^J$) and dashed curves reflect updates to the out-of-class prototype vector ($\mathbf{w}^K$).

to replace the current window defined by G(R)LVQ, rather add additional restriction such that prototypes are influenced by samples lying in some region about $\mathbf{M}^{JK}$. We consider the following additional windowing schemes a hypercube window, and a multi-dimensional Gaussian window.

Hypercube window: In order to restrict the development of the decision boundary to samples lying in a region about $\mathbf{M}^{JK}$, we define a hypercube window with width D which is the percentage of the distance between prototype vectors $\sqrt{d^{JK}}$ centered at $\mathbf{M}^{JK}$ for which updates may occur. If the input sample falls entirely within the window, then both the in-class and out-of-class prototypes are updated, otherwise no update occurs. We can capture

the function of the hypercube window with the indicator function

$$\mathbf{1}_D(\mathbf{x}^m) = \begin{cases} 1 \quad \text{if } (\mathbf{M}_i^{JK} - D\sqrt{d^{JK}}/2) \leq x_i^m \leq (\mathbf{M}_i^{JK} + D\sqrt{d^{JK}}/2) \\ \quad \forall i \in \{1, 2, \ldots, n\} \\ 0 \quad \text{Otherwise.} \end{cases} \tag{2.18}$$

We can write two new update functions $g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D)$ and $g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D)$ as:

$$g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D) = g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)\mathbf{1}_D(\mathbf{x}^m) \tag{2.19}$$

$$g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D) = g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)\mathbf{1}_D(\mathbf{x}^m) \tag{2.20}$$

where $g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)$ and $g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)$ are defined in Eq. 2.16 and Eq. 2.17 respectively. Fig. 2.6a and Fig. 2.6b illustrate the GRLVQ window overlaid with the hypercube window in one-dimension. Fig. 2.6c and Fig. 2.6d graphically illustrate the new update rule which is influenced only by the samples between the prototypes. The in-class (out-of-class) prototype vector is influenced more by samples slightly off-center when a correct (incorrect) classification occurs.

Multivariate Gaussian window: The hypercube window was a simple "0-1" window where either an update occurs or it does not. A different approach is to define a continuous function about $\mathbf{M}^{JK}$. The motivation is that a continuous window makes the transition between update and no update of the prototype vectors and relevance factors smooth, which may help reduce the oscillatory effects of the classification accuracy exhibited by GRLVQ. Based on the belief that we must refine decision boundaries by placing more emphasis on data points in the vicinity of $\mathbf{M}^{JK}$, we considered a multivariate Gaussian window. The mean vector of the Gaussian is simply $\mathbf{M}^{JK}$ and the variance is related to the percentage (D) of the distance $(\sqrt{d^{JK}})$ between prototypes $\mathbf{w}^J$ and $\mathbf{w}^K$:

$$\Sigma \triangleq D \times d^{JK}\mathbf{I}_{n\times n} \tag{2.21}$$

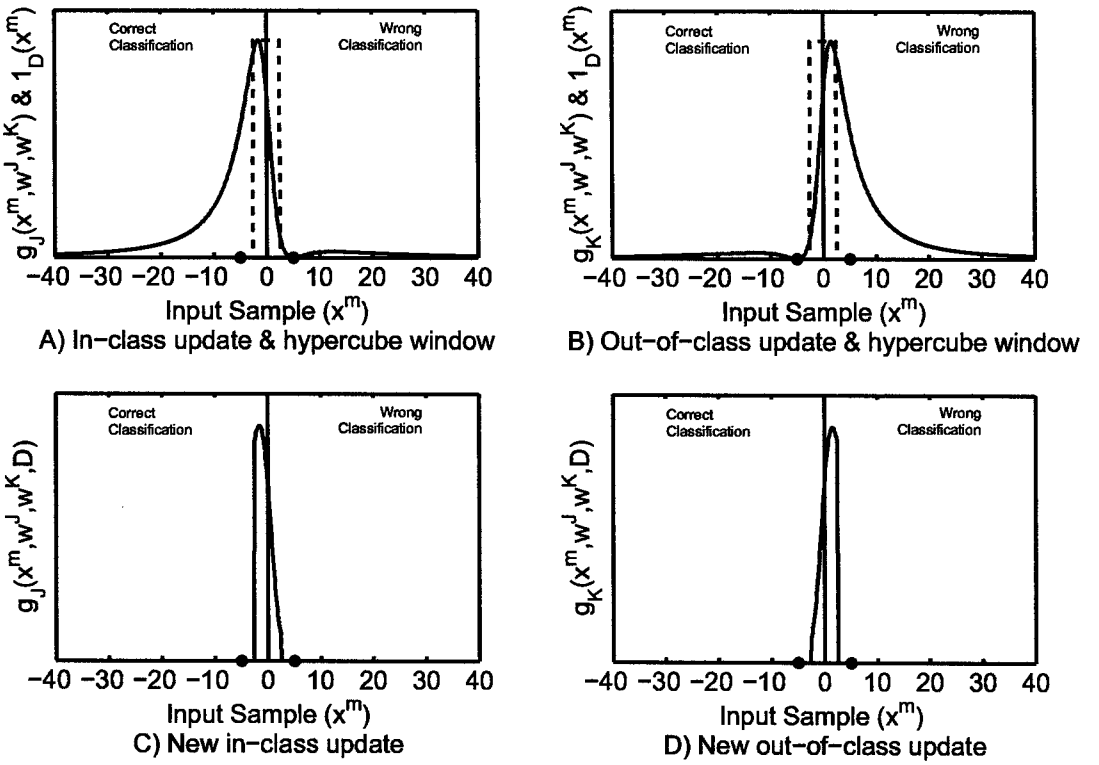


Figure 2.6 : This figure shows the application of the hypercube window (dashed) to the G(R)LVQ update (solid) with window $W = 0.4$ (40% of the distance between the in-class and out-of-class prototype vectors located at $\mathbf{M}^{JK}$) for a) G(R)LVQ update overlaid with hypercube window for the in-class prototype, b) G(R)LVQ update overlaid with hypercube window for the out-of-class prototype, c) effective in-class prototype update, and d) effective out-of-class prototype update.

where $I_{n\times n}$ is the $n \times n$ identity matrix. Using a non-scaled multivariate Gaussian (i.e., a Gaussian without the $((2\pi)^n \Sigma)^{-1/2}$ multiplicative term) is desirable so that the peak at $\mathbf{M}^{JK}$ evaluates to one. We can write two new update functions $g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D)$ and $g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D)$ as:

$$g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D) = g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K) \left(\exp^{-\frac{1}{2}[\mathbf{x}^m - \mathbf{M}^{JK}]\Sigma^{-1}[\mathbf{x}^m - \mathbf{M}^{JK}]^T}\right) \quad (2.22)$$

$$g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, D) = g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K) \left(\exp^{-\frac{1}{2}[\mathbf{x}^m - \mathbf{M}^{JK}]\Sigma^{-1}[\mathbf{x}^m - \mathbf{M}^{JK}]^T}\right) \quad (2.23)$$

where $g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)$ and $g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K)$ are defined in Eq. 2.16 and Eq. 2.17 respectively, $\mathbf{M}^{JK}$ is the midpoint between $\mathbf{w}^J$ and $\mathbf{w}^K$, and Σ is defined in Eq. 2.21. Fig. 2.7a and Fig. 2.7b illustrate the GRLVQ windows overlaid with the Gaussian window in one-

dimension. The continuous Gaussian window "refocuses" the emphasis on the midpoint between $\mathbf{w}^J$ and $\mathbf{w}^K$ as indicated in Fig. 2.7c and Fig. 2.7d. We are essentially morphing the bi-modal window to a uni-modal window without making the window degenerative (see Section 2.3.2.1).

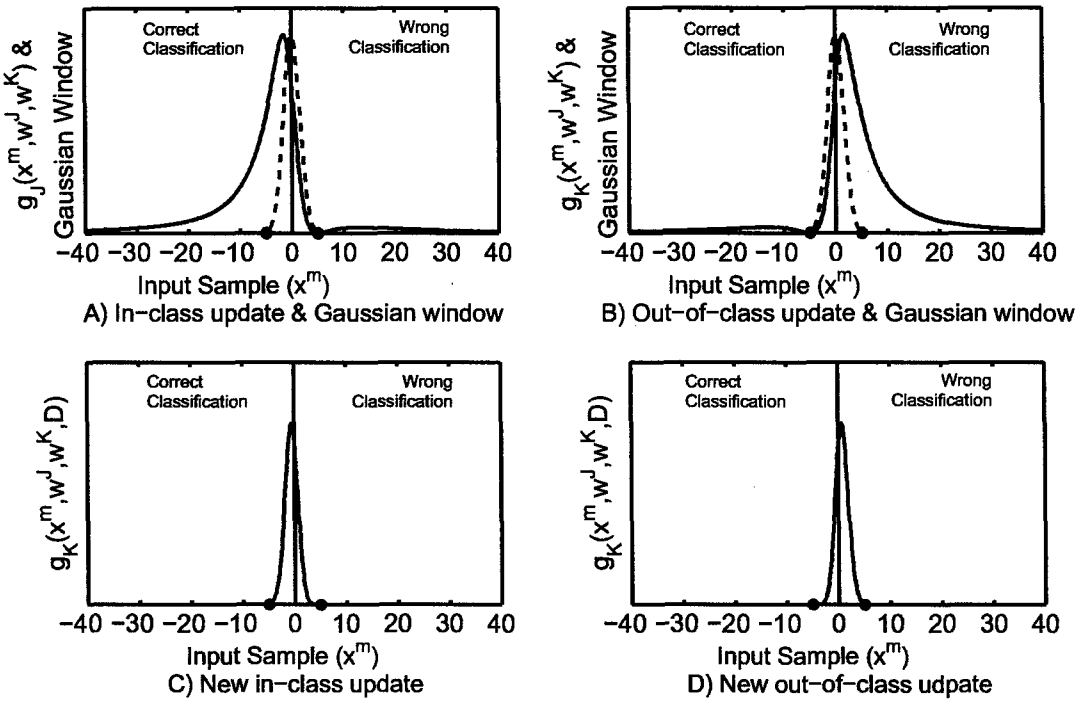


Figure 2.7 : This figure shows the application of the Gaussian window (with $D - 0.4$) to the GRLVQ update: a) G(R)LVQ update overlaid with Gaussian window for the in-class prototype, b) G(R)LVQ update overlaid with Gaussian window for the out-of-class prototype, c) effective in-class prototype update, and d) effective out-of-class prototype update.

2.3.1.4 Discussion of windowing and the effect on boundary definition

The windowing methods described above further restrict the window currently defined by G(R)LVQ. However, based on the results of our testing (not presented in the thesis), they do not offer any further improvement in classification accuracy and the window currently defined by the GRLVQ update rules performs as well or better. It does not appear necessary, or even useful, to further restrict the updates to the in-class and out-of-class prototype vectors

to samples around $\mathbf{M}^{JK}$. We conclude that the strict focus on the midpoint between $\mathbf{w}^J$ and $\mathbf{w}^K$ is not necessary to promote good class boundary definition. In analyzing the windowing effects of G(R)LVQ, we find two critical issue emerge, the potential for diverging prototype vectors and poor prototype utilization, which we discuss in Part II of this section below.

2.3.2 Part II – observations of GRLVQ

Although there does not appear to be a need to focus on updating the prototypes if the sample falls around their midpoint, we find two issues that can have a grave effect on classifier performance. First, we discuss the potential for diverging prototype vectors based on differential shifting. Second, we find that a large percentage of prototype vectors do not learn during the training process.

2.3.2.1 Potential for prototype divergence

Sections 2.3.1.1 and 2.3.1.2 show that prototype vectors are updated for a wide range of input samples. Our two windowing functions, $g_J(\cdot)$ and $g_K(\cdot)$, with the addition of a time decay term (τ) are:

$$g_J(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, \tau) = \frac{4d^K f'|_{u(\mathbf{x}^m),\tau}\Lambda}{(d^J + d^K)^2}, \tag{2.24}$$

$$g_K(\mathbf{x}^m, \mathbf{w}^J, \mathbf{w}^K, \tau) = \frac{4d^J f'|_{u(\mathbf{x}^m),\tau}\Lambda}{(d^J + d^K)^2}. \tag{2.25}$$

where $f'|_{u(\mathbf{x}^m),\tau}$ is defined as:

$$\begin{aligned} f'(\mu(\mathbf{x}^m),\tau) &= f(\mu(\mathbf{x}^m),\tau)\left[1 - f(\mu(\mathbf{x}^m),\tau)\right], \\ &= \frac{e^{\left(-\tau\frac{d^J-d^K}{d^J+d^K}\right)}}{\left(1+e^{\left(-\tau\frac{d^J-d^K}{d^J+d^K}\right)}\right)^2}. \end{aligned} \tag{2.26}$$

Increasing values of τ will cause the update window to converge from a bi-modal to a uni-modal window about $\mathbf{M}^{JK}$ (Fig. 2.8). In the limit as $\tau \to \infty$, $g_J(\cdot)$ and $g_K(\cdot)$ converge

to degenerative Gaussian functions (i.e., a Gaussian with infinitely small variance or a δ function). In this extreme case, the update window would consist of a single point located at $\mathbf{M}^{JK}$.

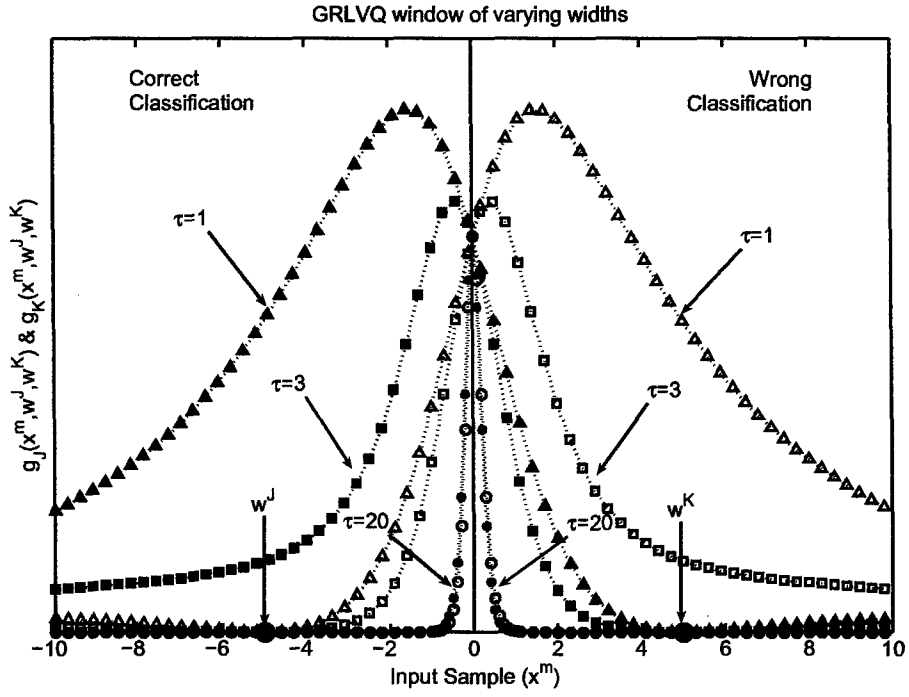


Figure 2.8 : The G(R)LVQ update window converges to a uni-modal window about the midpoint of $\mathbf{w}^J$ and $\mathbf{w}^K$ if a time decay factor τ is applied in Eq. 2.26. The filled triangle, solid square, and solid dot curves are for $g_K(\cdot)$ evaluated at $\tau = 1, 3$, and 20, respectively. The open triangle, open square, and open circle curves are for $g_J(\cdot)$ evaluated at $\tau = 1, 3$, and 20 respectively.

Sato and Yamada [17] suggest using a constant learn rate, starting with large windows and shrinking with increased training time. It is unclear how best to employ a time-varying decay term and careful consideration must be given so as not to hinder classifier development. Even though Sato and Yamada do not feel a decaying learn rate is necessary, it is still a good idea in order to control how far a prototype can be adjusted once the boundary is well defined. An indicator of a "well defined" boundary is the monitoring of the train and test classification history.

Properly decaying the sigmoid function is critical to ensure that divergence of the pro-

totype vectors does not occur. If the update window is left too wide for too long, then divergence can become an issue. To illustrate the possibility of divergence, we consider a simple two-class problem consisting of classes *C1* with a large number of samples and class *C2* with relatively few samples (depicted in Fig. 2.9). In our example, class *C2* will possess the nearest prototype(s) for class *C1*. Class *C1* has more samples than class *C2*, so prototype(s) from class *C2* will be pushed away from the boundary. Because class *C2* has fewer samples, it will unsuccessfully attempt to reposition its prototype(s) to redefine the boundary. Given this scenario, it is clear that divergence can occur. At the very least, the current GRLVQ update strategy can cause increased training time because prototypes properly positioned may be moved unnecessarily.

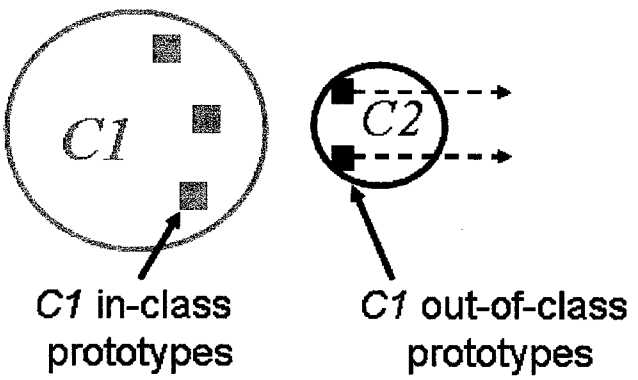


Figure 2.9 : An illustration of the divergence potential for a two-class problem. Class $C1$ has far more training samples than class $C2$ as indicated by the diameter of the circle. Prototypes in class $C2$ are the out-of-class prototype vectors of class $C1$ causing a disproportionate number of out-of-class updates for $C1$s prototype vectors than in-class updates, thus the divergence of $C2$s prototype vectors can occur.

2.3.2.2 Prototypes that never learn

During our investigations of GRLVQ, we found that many of the prototype vectors never learn during the training process. This should come as no surprise since it is a common problem amongst prototype-based learning algorithms. For example in the case of the 23-class hyperspectral classification problem discussed in detail in Section 3.2, we find that only

60 out of 115 ($\approx$ 52%) of the prototype vectors learn during classifier training (this means that only 52% of the prototypes were different from their initial random state at the end of training). For the 35-class hyperspectral classification problem, this issue is much more severe. Here, only 67 of 175 ($\approx$ 38%) learn during the training process. As the geometric relationships between the classes in a given problem becomes more difficult, the more likely this problem is to have a serious negative effect on the classification results.

2.4 Our contribution – Generalized Relevance Learning Vector Quantization Improved (GRLVQI)

In our analysis of GRLVQ, we find it suffers from two problems: the potential for diverging prototype vectors and poor prototype utilization. The divergence problem is addressed by changing the learning rule discussed in Section 2.4.1. We address the poor prototype utilization by incorporating DeSieno's conscience mechanism [19] for in-class prototype selection discussed in Section 2.4.2. This equiprobabilistic (maximum entropy) solution ensures all prototypes receive an opportunity to learn during the training process. Our improvements to GRLVQ (dubbed GRLVQI) result in decreased training time and better classification.

2.4.1 Addressing the problem of prototype divergence

Our in-class conditional update prevents the potential for divergence by changing the update strategy for correctly classified samples. The original GRLVQ update the winning out-of-class prototype vector unconditionally, which can cause divergence as discussed in Section 2.3.2.1. We argue that it is only necessary to adjust the out-of-class prototype if a sample is incorrectly classified. The new update rule is then to move the in-class prototype towards the sample and the out-of-class prototype away from the sample only if a sample is misclassified. If the sample is classified correctly, we adjust the in-class prototype towards the sample and leave the out-of-class prototype unchanged. We can view the effect of our

update rule in Fig. 2.10 where we see the original GRLVQ update occurs for an incorrect decision (indicated with the solid curve). When a sample is classified correctly, only the in-class prototype vector is updated (indicated by the dashed curve of Fig. 2.10).

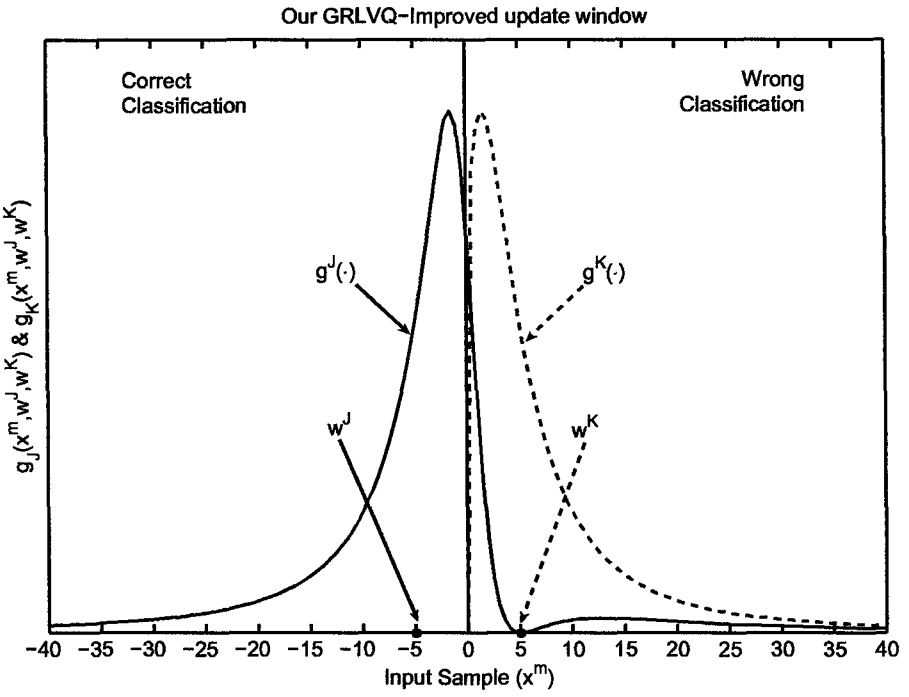


Figure 2.10 : Effective update window for our improved GRLVQI. The values of $g_J(\cdot)$ and $g_K(\cdot)$ along the y axis are evaluated for fixed prototype locations at -5 ($\mathbf{w}^J$) and 5 ($\mathbf{w}^K$) where the input sample $\mathbf{x}^m$ varies along the x axis. Solid curves reflect updates to the in-class prototype vector ($\mathbf{w}^J$) and dashed curves reflect updates to the out-of-class prototype vector ($\mathbf{w}^K$).

2.4.2 Improving prototype utilization

Poor prototype utilization is a classic problem with prototype-based learning methods, with good solutions for unsupervised learning algorithms [19, 20]. However, solutions for supervised learning algorithms have not been addressed in the literature. DeSieno's conscience mechanism [19] biases the Euclidean distance between the input sample $\mathbf{x}^m$ and the prototype vector $\mathbf{w}^p$, which modifies the chance of the prototype vector $\mathbf{w}^p$ becoming the winner. The bias is calculated from the wining history for each prototype $\mathbf{w}^p$ so as to discourage frequent

winners from winning more often and encourage the selection of infrequent winners.

Updates to the frequency F^p for the winning prototype vector $\mathbf{w}^p$ (this prototype is the in-class winner $\mathbf{w}^J$) is:

$$F^p_{\text{new}} = F^p_{\text{old}} + \beta\,(1.0 - F^p_{\text{old}})\,. \tag{2.27}$$

For the remaining prototypes, the frequency is adjusted as:

$$F^p_{\text{new}} = F^p_{\text{old}} + \beta\,(0.0 - F^p_{\text{old}})\,. \tag{2.28}$$

The β term in Eq. 2.27 and Eq. 2.28 is a user-defined parameter that controls the amount of update to the frequencies F^p.

For winner selection, one uses the biased Euclidean distance between the input sample $\mathbf{x}^m$ and the prototype $\mathbf{w}^p$:

$$d_{\text{Bias}} = d^p - B^p, \tag{2.29}$$

where B^p is defined as:

$$B^p = \gamma\left(\frac{1}{P} - F^p_{\text{old}}\right). \tag{2.30}$$

The γ term in Eq. 2.30 is a user-defined parameter that controls the amount of bias applied to the Euclidean distance. Conscience learning is an equiprobabilistic winner selection strategy which allows for an optimal maximum entropy vector quantization and allows prototype vectors to contribute quickly [19].

We can take advantage of DeSieno's conscience mechanism for GRLVQI by applying a separate conscience for each class. That is, during in-class prototype selection, we bias the Euclidean distance as described by Eq. 2.29 where the bias is defined in Eq. 2.30. We then update the frequency of the winning in-class prototype vector $\mathbf{w}^J$ as in Eq. 2.27 and the frequency of the non-winning in-class prototype vector as in Eq. 2.28. During the out-of-class

winner selection, we *do not* bias the Euclidean distance. Further, the unmodified Euclidean distance is used in the update rules for $\mathbf{w}^J$, $\mathbf{w}^K$, and λ.

2.5 Discussion

This chapter promoted and laid out a framework for joint feature extraction and classification models as the right choice for ensuring a set of features that maintain the discrimination capability of the data. We investigate the neural paradigm GRLVQ for joint classification and feature extraction of remotely sensed hyperspectral data. The GRLVQ gives us a convenient way of reducing the dimensionality of the data based on rank-ordering the relevances (or importance) placed on each of the input dimensions during learning. Keeping only the N input dimensions with the corresponding N largest relevances will give us the best combination of input dimensions which will yield the best possible classification (using GRLVQ as the classifier). An original analysis of the LVQ2.1 and GRLVQ windowing methods was accomplished where we demonstrated that restricting updates of prototype vectors to samples lying in the mid-region of the in-class and out-of-class winning prototype vectors dispelled a common belief that this additional restriction is necessary. Further analysis and experimentation with GRLVQ revealed two deficiencies which we addressed as a contribution to GRLVQ yielding our GRLVQ-Improved (GRLVQI). The first deficiency (unconditional updates to the out-of-class prototype vectors) leads to the potential for prototype divergence. The second deficiency is poor prototype utilization, which results in an underdeveloped classifier that is incapable of realizing its true potential. The end result of our GRLVQI is the elimination of the potential for prototype divergence by changing the prototype update rules and an increase in classification accuracy by implementing a maximum entropy solution for in-class prototype winner selection (Hammer and Villmann also noted that a maximum entropy solution could bring improvements to GRLVQ [1]).

In the next three chapters, we demonstrate the improved performance of GRLVQI over GRLVQ in a principled and systematic fashion. We start by defining three challenging hyperspectral classification problems. Using these three problems, we compare the classification accuracy and feature extraction results achieved by GRLVQ(I). We then expand the analysis of our classification results to show that the improvements of GRLVQI over GRLVQ are significant.

Chapter 3

Test and evaluation framework for comparing GRLVQ and GRLVQI

Comparing the performance difference between two similar classifiers is an involved task. Consideration must be given as to the type of data one evaluates to meet the intended use of the classifier. Additional care must be taken to design tests such that they are unbiased and as "independent as possible". Depending on the type of data one evaluates, repositories such as the UCI machine learning repository [21], provide a good source of different predefined problems with well studied labeled training samples. Since our research focuses on the analysis of hyperspectral data, we are only interested with the performance difference of GRLVQI over GRLVQ on data sets for which GRLVQI can have an advantage over GRLVQ. For relatively simple data sets, such as those found in the UCI machine learning repository, our GRLVQI likely does not offer performance improvements. Unfortunately, repositories similar to the UCI do not exist for hyperspectral data or other types of data with similar complexity and dimensionality making it difficult to find a variety of well studied test cases to evaluate GRLVQ(I).

In the first part of this chapter, we provide background information on hyperspectral images. Next, we present the Lunar Crater Volcanic Field (LCVF), NV remotely sensed hyperspectral image, which is a sufficiently complex and high-dimensional data set that will give a good evaluation of the classification and feature extraction capabilities of GRLVQ(I). Hyperspectral images also represent a large family of data that are widely pursued for sophisticated information in many areas of science, engineering, industry, and decision making functions. To achieve good classifications, we must take care in designing our GRLVQ(I)

networks. We present the network configuration and the schedule of time decayed learning parameters (e.g., learn rates and conscience parameters where appropriate) in light of design guidance from the literature so we may have the best possible GRLVQ(I) networks for analyzing the LCVF scene. Finally, we present a challenging baseline classification of the LCVF data set (from a previous study in [22]) to which we can compare the results of GRLVQ(I).

3.1 Background – hyperspectral images

Hyperspectral images are complex high-dimensional data sets used in many facets of science, engineering, and industry, often serving as decision making tools. The users of this sophisticated imagery require the very best in classification accuracy. We aim to provide neural solutions to address the needs of the community. We do this by demonstrating the effectiveness of the joint classification and feature extraction capabilities of GRLVQ(I) on hyperspectral data. We provide a description of hyperspectral data in this section.

Hyperspectral images are a collection of (potentially) hundreds of co-registered images where each image records the measured light response in a narrow frequency band, at every pixel. This collection of images is represented conceptually as an image cube (Fig. 3.1 left). Each pixel in the image cube has associated with it an n-dimensional vector, called spectrum (Fig. 3.1 right), the elements of which are the measured light intensities of the respective wavelengths at that pixel location.

Hyperspectral images provide one with a wealth of information. The spectral resolution of typical hyperspectral sensors allows one to discriminate nearly any material, which is why hyperspectral images are used in a wide range of applications, including both terrestrial and extra-terrestrial studies. See, e.g., Howell et al. [24] for a study on asteroid types. Hunt and Salisbury [25, 26] and Hunt et al. [27–36] have a series of seminal papers on a spectroscopic study of rocks and mineral identification based on absorptions in the visible

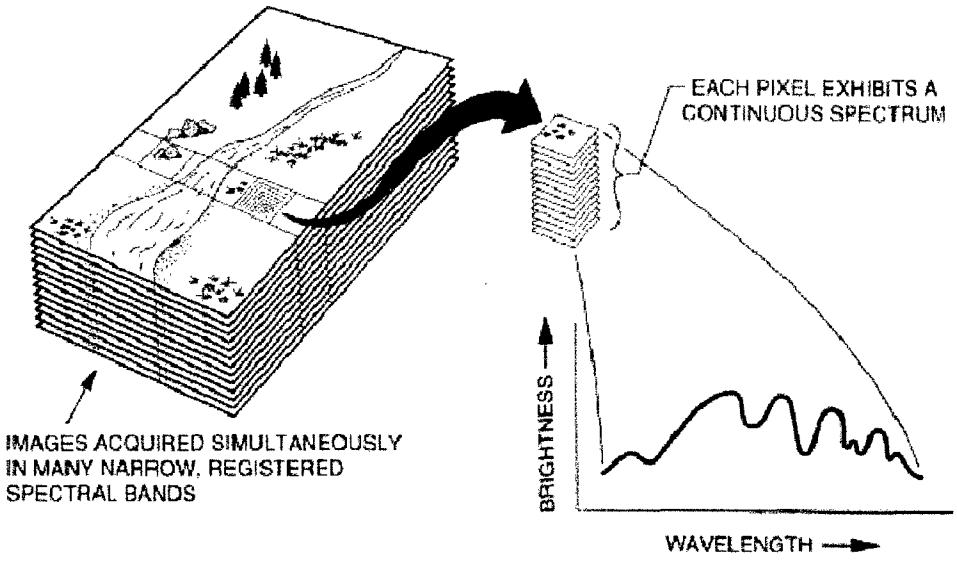


Figure 3.1 : A hyperspectral image cube where each pixel is an n-dimensional vector which is the spectrum of the material in that pixel. Figure from Campbell [23].

(0.3-0.7 μm) and near-infrared (0.6-2.6 μm) spectral regions. This series of papers provides a nice discussion on how the *electronic* and *vibrational* processes interact to create the spectral features we rely on for classification. These spectral features are called absorptions and are based on the repeatable physical process of the interaction of light with materials at different wavelengths.

Hyperspectral images not only allow us to identify the mineralogical composition of materials, one can make inferences on other material characteristics. For example, differences in temperature manifest in spectra makes temperature inference possible (see, e.g., Roush and Singer [37]). Further, the size of the particulate matter is also known to affect the spectral response [38] and therefore grain size inferences are also possible.

3.2 The Lunar Crater Volcanic Field (LCVF) data set

We use training and testing samples from a remotely sensed hyperspectral image of the Lunar Crater Volcanic Field (LCVF), NV test site acquired by the NASA/JPL AVIRIS

hyperspectral sensor [39] in 1994. A natural color composite image of the LCVF sub-scene is shown in Fig. 3.2. It is a 614 × 420 pixel image with 194 spectral bands after deleting the saturated atmospheric water bands containing irrecoverable data. We normalize each spectrum (194-dimensional vector) by its l_2 norm in order to cancel linear effects [40], such as shading caused by viewing geometry. This normalization makes spectral classes more uniform, which in-turn aides the classification process. One undesirable effect of spectral normalization is that it removes the differences in (geometric) albedo, i.e., erases all of the real differences between materials that have the same spectral signature and only differ in their albedo [40]. Although this problem is rarely encountered, one must be aware of the possibility so that if it does occur, post-classification processing can be done to separate the affected materials.

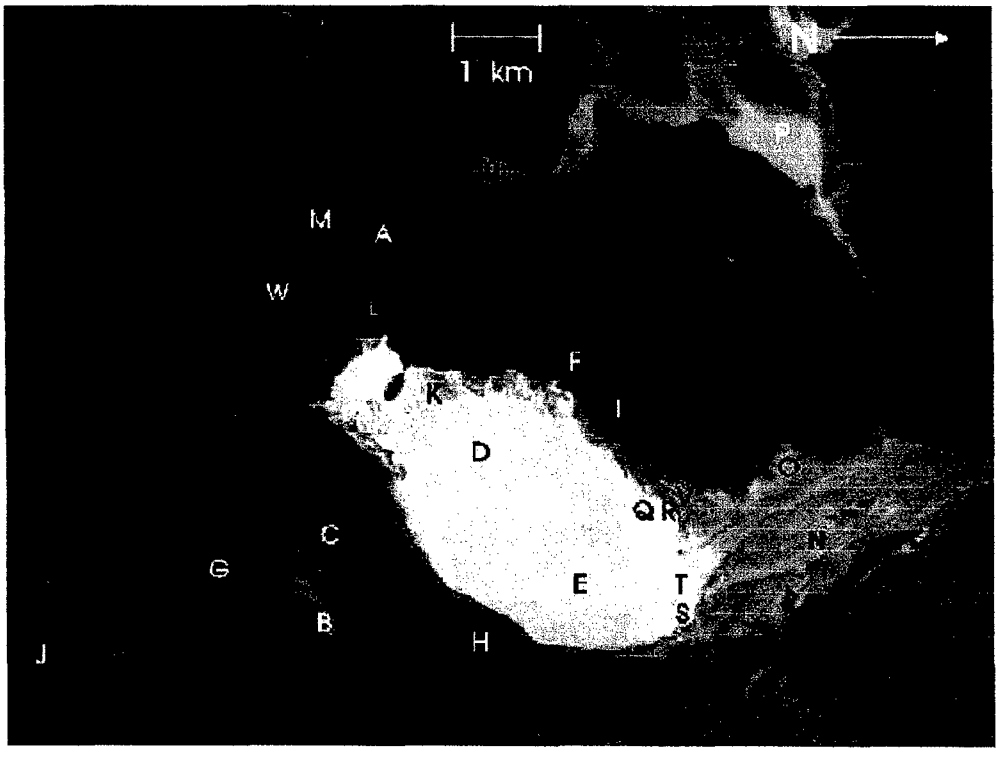


Figure 3.2 : Natural color composite image of the Lunar Crater Volcanic Field (LCVF), NV scene obtained by the NASA/JPL AVIRIS imager in 1994. Locations of 23 surface cover types are indicated by labels (see Table 3.1 for a description of the material labels shown here). Additional details and previous analysis of the LCVF image are available in [22].

We use three classification problems of increasing difficulty to demonstrate the improvements of our GRLVQI over GRLVQ. First, we evaluate a 7-class problem. Second, we evaluate a 23-class problem for which a description of the classes is listed in Table 3.1 along with the class labels and the number of training samples. The 7-class problem is a subset of the 23-class problem consisting of classes **A**, **D**, **E**, **H**, **I**, **L**, and **W**. Further details and past analysis on the 23-class data set are available in [22]. Finally, we evaluate a more difficult 35-class problem, an extension of the original 23-class problem. The additional twelve classes were discovered via independent Self-Organizing Map (SOM) clustering [22] of the LCVF scene after careful scrutiny of their statistics and spatial distribution. The 35-class problem adds the following 12 classes: **X-Z**, **a-e**, and **g-j**. Class labels and corresponding mean spectra for all 35 classes is provided in Fig. 3.3.

Class	Cover description	#	Class	Cover description	#
A	Hematite-rich cinders	72	M	Alluvium #3 (iron rich)	14
B	Rhyolite of Big Sand Spring Valley	22	N	Dry wash #1	15
C	Alluvium #1	50	O	Dry wash #2	54
D	Dry playa	160	P	Dry wash #3	45
E	Wet playa #1	115	Q	Wet playa #2	15
F	Young basalt	21	R	Wet playa #3	14
G	Shingle Pass tuff	7	S	Wet playa #4	15
H	Alluvium #2 (with mixed scrub, brush, rocks, and soil)	50	T	Wet playa #5	18
I	Old basalt	36	U	Alluvium #4 (also	
			U	iron rich)	36
J	Dense scrub brush stands	12	V	Wet playa #6	12
K	Basalt cobbles on playa	37	W	Ejecta blankets #2 (primarily unoxidized cinders with smaller percentage of hematite-rich cinders)	33
L	Ejecta blankets #1 (mixed hematite-rich and unoxidized cinders)	78			
				Total #	931

Table 3.1 : Class descriptions, labels, and number of samples for the 23-class problem. The 7-class problem is the following subset of classes: **A**, **D**, **E**, **H**, **I**, **L**, and **W**.

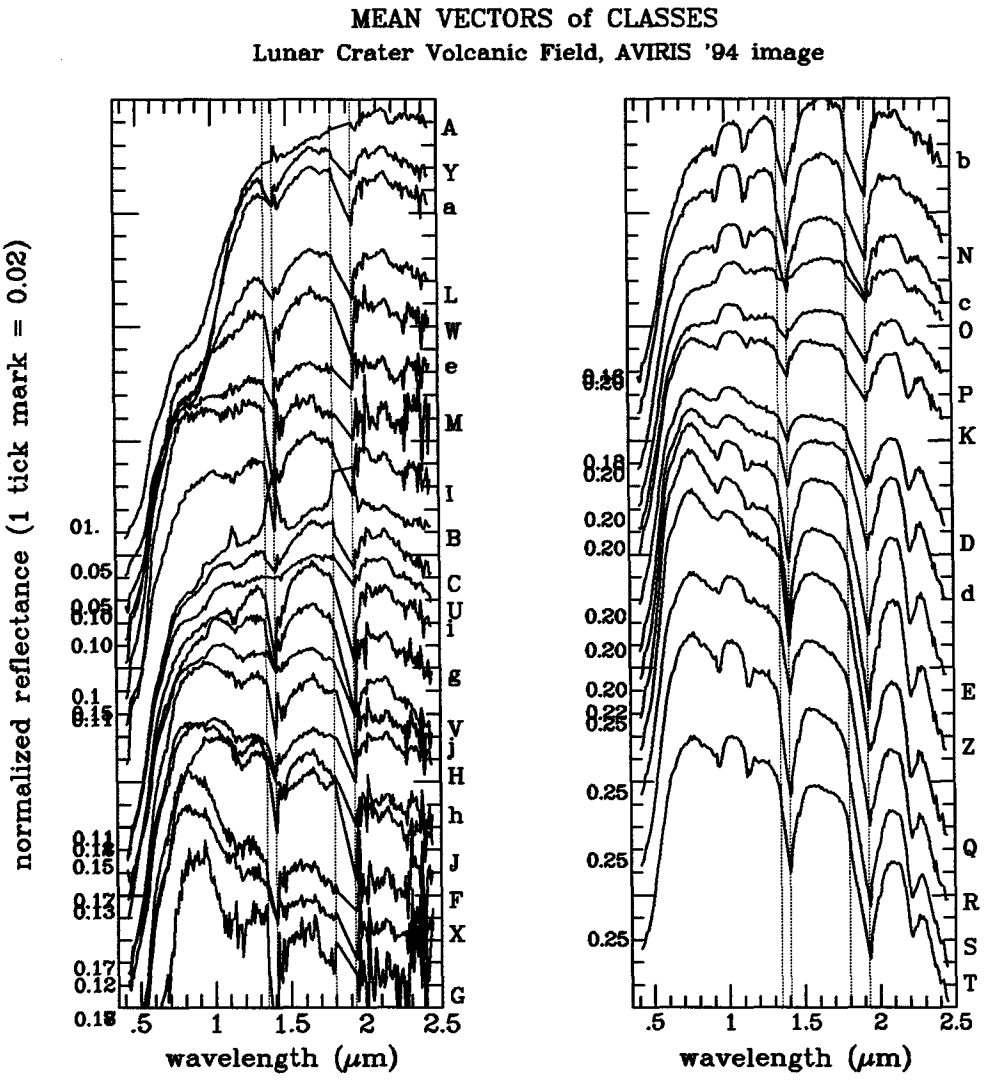


Figure 3.3 : Average spectra of the original 23-class problem (classes **A-W**) and the 12 additional classes (**X-Z**, **a-e**, and **g-j**) for the 35-classes problem.

3.3 A benchmark classification of the LCVF data set

To assess the classification performance of GRLVQ(I), we compare the results against a baseline classification of the 23-class problem with a hybrid artificial neural network (ANN) using all 194 spectral features. The hybrid ANN is capable of exploiting the intricacies of high-dimensional giving us more challenging classification benchmark. Classification accuracy achieved by the hybrid ANN is 92.1% averaged over the three independent jack-knife runs. We believe the classification accuracy of the hybrid ANN on the testing data to rep-

resentative of its true performance capability. The same hybrid ANN architecture was used in previous studies to classify the entire LCVF AVIRIS scene where the data presented in Table 3.1 served as the training spectra for the 23-class problem ([13,41]). The assessment was evaluated according to rigorous statistical assessment based on sampling theories which included a large number of "ground truth" pixels. The classification accuracy of the hybrid ANN classifier was $\approx$ 90%, followed by the MED classifier with 83%, a Spectral Angle Mapper classifier with $\approx$ 80%. The Maximum Likelihood (ML) classifier could not be applied to the 194-dimensional data for lack of sufficient number of training samples, which is dimension-dependent for the ML.

The accuracy produced by the hybrid ANN on the entire AVIRIS image [22] is lower than that produced on the 23-class data set used in this study. This is simply because we are using clean spectral specimens from the LCVF image for both training and testing, where in the original study all of the noisy pixels participated in the classification. Further, the previous studies were conducted during a time when computing power was more limited than what is available today. As such, it may be possible that with longer training the hybrid ANN would achieve somewhat higher classification accuracy.

The hybrid ANN used for our benchmark classification of the LCVF data set is described in several earlier publications (see, e.g., Merényi [22], and references therein). It consists of a 2-dimensional Self-Organizing Map [2] (SOM) as a hidden layer and a categorizational output layer that learns using the Widrow-Hoff learning rule. In the first part of the training process, the SOM is allowed to learn in unsupervised mode where it learns the structure of the data. After a certain number of training steps have passed, the supervised learning of the output layer is turned on. This combination of unsupervised and supervised learning is powerful since the preformed clusters in the SOM help the output layer refuse the learning of inconsistent class labels resulting in an overall better classification of the data.

3.4 Design of GRLVQ(I) classification experiments

We consider classification problems with varying degrees of difficulty (described in the previous section) to demonstrate the effectiveness of GRLVQI over GRLVQ. We will use the notation GRLVQ(I) to indicate that we are talking about both GRLVQI and GRLVQ. The purpose of the relatively easy 7-class problem is to demonstrate the effect of the in-class conditional update on the classifier's rate of convergence. Here, we *do not* use conscience learning so we may isolate the effect of our in-class conditional update rule. Next, we execute GRLVQ(I) on the 23-class data set to show improved classification accuracy due to the in-class conditional update and conscience learning together. Evaluation of this 23-class problem is especially important since there is previous analysis on this exact problem using neural paradigms which will serve as a performance baseline for evaluating the effectiveness of GRLVQ(I) for classifying hyperspectral images. Finally, we evaluate the classification performance of GRLVQ(I) on the 35-class data set to further demonstrate the advantages of GRLVQI over GRLVQ.

A k-fold cross validation is a commonly accepted practice for classifier evaluation in order to reduce the bias and variance of the estimated prediction error [42]. The practice is to randomly select a fraction $(\frac{k-1}{k})$ of the known labeled training samples for training the classifier and the remaining $(\frac{1}{k})$ for evaluating the classifier [42, 43]. A total of k tests are performed, using a different set of testing and training samples for each run. Specific guidance on choosing the value of k is not universal. Haykin [43] suggests a value of $k = 5$ while Hastie et al. suggest $k = 5$ or $k = 10$ [42]. It is recognized that the value of k often depends on the problem at hand. Since our classification problems have a large number of samples and a large number of classes and computationally intensive in the current software implementation, we use a 3-fold cross validation. Two-thirds of the known labeled samples are

used for training and one-third for testing (Table 3.2). When reporting overall classification results for a given problem (e.g., 7-class, 23-class, or 35-class), we average the test results from each of the three independent jack-knife runs.

	7-Class	23-Class	35-Class
Training Samples	347	621	976
Test Samples	173	310	488
Total Samples	520	931	1464

Table 3.2 : Number of training and testing samples for the 7-class, 23-class, and 35-class problems.

3.5 GRLVQ(I) classifier design

Designing an LVQ-type classifier for a particular classification problem can be successfully accomplished in four steps. First, one must decide on the number of classes. Second, one must decide on the number of prototype vectors to assign to each class. One may assign the same number of prototypes for each class, or based on a priori knowledge, assign an appropriate number of prototype vectors which may be different for each class. Third, prototype vectors must be initialized. Finally, classifier parameters must be chosen.

3.5.1 Assigning prototypes to classes

The number of prototype vectors assigned to each class can have a significant impact on the classifier's ability to learn class boundaries [44]. We would like to turn to existing theory for guidance on how best to approach this design consideration. Generalization bounds based on Vapnik-Chervonenkis (VC) dimension [14] (see e.g., [45]) and more recent works based on Bartlett and Mendelson's [46] Gaussian complexity [47], only give us guidance on the total number of prototype vectors in our classifier, not on the distribution of those prototypes. Unfortunately, this existing theory for generalization bounds for LVQ-type classifiers are of little use here. This leaves us to design our networks based on "rules-of-thumb" or heuristics.

In our problem, we use five prototypes per class for each of the three classification problems. This number was empirically determined and shows good results for our experiments. There is no reason to assume that each class should have the same number of prototype vectors. It is possible that varying the number of prototypes across classes based on class characteristics could produce improved results, or perhaps reduce computation time by ensuring a minimal set of prototypes are used. The size of the class, its geometric relationship with other classes, and modality of the class distribution, can each influence how one should assign the number of prototypes per class.

Using a self-organizing map (SOM) in a preprocessing phase to derive an ideal distribution of the number of prototypes for each class may be a more principled approach. We could apply labeled training samples to prototype vectors in the SOM and observe the number of prototype vectors assigned to each class label. This observed number can be used directly, or in proportion, as the number of prototypes to assign to each class in the GRLVQ(I) network.

We do not focus our efforts on solving this issue as our initial objective was to see improved classification performance. Our results indicate that our simplistic approach of assigning the number of prototypes vectors for each class shows good results. The above is a more principled approach to determine the number of prototype vectors to assign to each class because we can potentially improve classification accuracy, decrease training time, or both, by using a minimal set of prototype vectors.

3.5.2 Initializing prototype vectors

The initial state of an LVQ's prototype vectors can affect the generalization ability of the classifier [2]. The literature addresses this problem in several ways. Prototype vectors can be initialized based on the sample distribution or based on the data extremes as in [48]. They can be moved from some initial random state to a state suitable for refinement [1,2]

by using some other algorithm. Or, prototypes can be randomly initialized and one can use methods to ensure prototypes converge to a local optimum [49]. We found that the addition of conscience learning to address poor prototype utilization allows us to use random initialization and obtain consistent results. That being the case, we do not employ fancy initialization schemes as they are not needed for our work. We scale the data on the interval $[0, 1]$ to correspond to a region on the sigmoid where the slope is large, which has the effect of speeding up convergence. By initializing the prototypes to the center of that interval (initial prototype values are drawn randomly from a uniform distribution on $[0.4, 0.6]$) for which the data are scaled, we further aid the convergence rate of the classifier. We use the same initialized state for each classification problem when testing both classifiers.

3.5.3 Choosing GRLVQ(I) learning parameters

There are no hard-and-fast rules, or much in the way of theoretical considerations when choosing learning parameters. One may consider this part of the process as the "art of classifier design". The parameters we present in this section result from a process of trail and error. During our trial and error, we found that the quality of the final classification was not overly sensitive to specific parameter values. An old lesson relearned was that one should decay parameters as learning progresses. Exactly this point brought more dramatic improvement on classification accuracy than the process of "parameter tweaking". We use a learning schedule instead of implementing a time-decay function for each of the learning parameters giving us better control over the learning process. This is described by the schedule employed for GRLVQ(I) learning in Table 3.3.

Table 3.3 lists the learning parameters for GRLVQ(I) and the conscience parameters for GRLVQI for the 23-class and 35-class problems. The parameters are decayed using a predetermined schedule based on the number of training steps. Although ideally all learning

parameters would decay with continued training, we found that a constant γ for conscience control worked well in our classification problems.

Schedule of GRLVQ(I) Learning Parameters

	GRLVQ(I) Params			Consc. Params	
Training Steps (TS)	ϵ^λ	ϵ^J	ϵ^K	γ	β
$0 < TS \leq 400K$	0.005	0.025	0.025	2	0.35
$400K < TS \leq 800K$	0.0025	0.0125	0.0125	2	0.3
$800K < TS \leq 1.2M$	0.001	0.005	0.005	2	0.225
$1.2M < TS \leq 1.6M$	0.0005	0.0025	0.0025	2	0.125

Table 3.3 : The GRLVQ(I) learn schedule for the 23-class and 35-class problems. The first three learn rate parameters are used for both GRLVQ and GRLVQI. The last two parameters control the conscience aspect of the learning and are for GRLVQI only. The number of training steps (TS) is rounded to the next thousand.

3.6 Discussion

This chapter laid the groundwork for the rigorous evaluation and comparison of GRLVQ(I). It introduced a hyperspectral data set that is sufficiently high-dimensional and complex to require more sophisticated classifiers than most currently used for meaningful high-quality classification, and will serve to validate the improvements of GRLVQI over GRLVQ (shown in Chapter 4). We use an independent evaluation of the same data to set a challenging benchmark classification for which any less accuracy achieved by GRLVQ(I) would invalidate the use of the GRLVQ(I) classifiers on high-dimensional complex data such as hyperspectral data. The design of our classification tests follow accepted standard practices and ensure a meaningful evaluation of classifier performance. The progressively more difficult classification problems we defined will ensure a fair comparison of the classification and feature extraction performance of GRLVQI and GRLVQ giving one a good indication as to how difficult the problem must become before GRLVQI is better suited than GRLVQ. Our network configuration was developed by carefully following four phases of LVQ network design while at the same time evaluating best-practices from the literature to ensure optimal performance.

Chapter 4

Application of GRLVQ and GRLVQI on the Lunar Crater Volcanic Field data set

Before comparing GRLVQ and GRLVQI, we first evaluate their classification and feature extraction success separately. In this chapter, we discuss important background information which details methodology for evaluating GRLVQ(I) performance. We present methodology for determining how many relevance-ranked features to retain and how we independently assess the discrimination capability of the retained feature set. We conclude with the presentation of the classification and feature extraction results for GRLVQ(I) processing on the three classification problems discussed in Chapter 3. Results are presented in light of the accuracy discussion in Section 4.1, feature retention methodology and independent assessment of GRLVQI features from Section 4.2.

4.1 Measuring classifier success

The current literature suggests that methods other than classification accuracy should be used to evaluate classifier performance ([50] and references therein). Demšar in [50] (and others) suggest using the area under the receiver operating characteristic (ROC) curve (AUC). Samples of the ROC generated along the ROC by setting a series of thresholds for the output of the classifier, classifying the data for each threshold, and tabulating the results. The tabulated results are used in methods such as the trapezoidal integration method [51] to estimate the AUC. The AUC is not without its problems. First, much of the current literature restricts the AUC method to binary classification problems [51–54]. Some extensions of the AUC exist for the multi-case (see Fawcett [55] for a nice summary of multi-class ROC and

AUC). One by Provost and Domingos [56] essentially create a two-class problem with the class currently under evaluation verses the rest of the classes. The AUC is then a combination of the weighted AUCs where the weighting is the frequency of the prevalence of the class under evaluation. In this case, we would require a total 23 (35) AUC estimates for the 23-class (35-class) problems. According to [55], the method described in [56] is sensitive to the distribution of the classes and error costs. Hand and Till [57] approach the problem differently and have a solution which is not sensitive to class distributions or the error cost [55]. Hand and Till's method measures the discrimination between pairs of classes, of which there are $\frac{N_c(N_c-1)}{2}$ where N_c is the number of classes in the classification problem. For the 23-class (35-class) problem, there are 293 (595) pairs of classes. For the number of classes we are interested in classifying, this method is cost prohibitive.

Second, according to Holte and Drummand [53], there is no agreed upon best-practice for averaging ROC results for a k-fold cross validation (we use a 3-fold cross validation). Finally, the interpretation of performance is not cut-and-dry. For example, the estimate of the AUC can serve as one performance indicator. A second indicator is when the ROC of classifier 1 dominates the ROC of classifier 2 [54, 58]. In cases where ROC-curves cross, the interpretation is that classifier one performs better than classifier two for different ranges of operating points [53]. Given the issues discussed above for the multi-class AUC, that there is no agreed upon method for combining multiple ROCs (used to find the AUCs), and that the interpretation of the results is not clear, we are unable to use the AUC for performance evaluation. For these reasons, we use classification accuracy.

Classification accuracy of each of the independent jack-knife tests can be calculated using one of two methods. The first method is the *non-equal weighted* accuracy, which is simply the total number of correctly classified samples divided by the total number of samples. This accuracy measure favors larger classes and suppresses the contribution of small classes. The

second method is the *equal weighted* accuracy, which is the average of the individual class accuracies. The *equal weighted* method is used here because it gives each class, regardless of its size, equal contribution in the evaluation of the classifier.

4.2 Background on feature retention and evaluation

There is an interdependence between the classification aspect of GRLVQI on the retention of the important input features through the relevance weighting of the input dimensions. An age-old problem with feature selection is deciding how many to keep. For GRLVQ(I) computed relevances, one natural thought is to select input dimensions in decreasing order of relevance, checking classification accuracy with the GRLVQ(I) classifier as each new feature is added. The number of features one retains is then drawn from the tabulation of results based on some criterion (e.g., maximum achieved accuracy or a the tradeoff between retained features and achieved accuracy). However, this method might be considered incestuous since the same classifier used in calculating the importance of the input dimensions would also be used to evaluate feature quality. Further, it says nothing about the universality of the feature set.

A more meaningful method of feature retention and evaluation is to use an independent classifier to select the number of features for the best tradeoff between the retained features and achieved classification accuracy. An independent classifier can also serve the purpose of evaluating the discrimination capability of the extracted (retained) features. We use the Minimum Euclidean Distance (MED) as our independent classifier in this thesis. To determine and evaluate the GRLVQ(I) features, we select the features in a cumulative fashion starting with the input dimension corresponding the largest GRLVQ(I)-computed relevance, and each additional input dimension is selected in descending order of GRLVQ(I)-computed relevance. The MED classification is performed with each newly added feature. This process

is continued until all features are used in the MED classification of the data.

We recognize that using the MED for feature retention and evaluation may not be the best approach. Philosophical differences between an LVQ classifier and the MED make GRLVQ(I) features suboptimal for the MED. An LVQ classifier uses multiple prototype vectors per class to define class boundaries. Prototype vectors learn local class structure (specifically the margins between adjacent classes), thus the computed relevances of features reflect more localized differences. In contrast, the MED is equivalent to an LVQ with a single prototype vector where that prototype is exactly the mean of the training data for each class. A more meaningful set of features for the MED would reflect the difference in the mean between neighboring classes.

Based on the above discussion, relevances computed by GRLVQ(I) may not identify a set of globally acceptable features which may be used in any classifier with the same classification accuracy success as in GRLVQ(I) itself. This in turn affects the reliability of the determination of the required number of features for peak classification accuracy. Regardless, the simplicity and speed of the MED make it an attractive method for feature selection and evaluation. Further, it serves our purposes of providing a coarse look at feature retention and class discrimination capability of those retained features. For this reason we chose it for an initial assessment in this thesis. The promising outcome encourages an investigation of more refined methods in future work.

In using relevance selected features in the MED classifier, one must decide whether or not to weight the data dimensions with the computed relevances. Although the relevance weighting could be applied to the data, in light of the philosophical differences between GRLVQ(I) related features and an MED-type feature, it makes more sense not to apply the relevance weighting. Further, in Chapter 6 we compare relevance selection of wavelet coefficients, largest magnitude wavelet coefficient selection, and relevance selection of spectral

features where it will make for a fair comparison of features if relevance is not applied to the data.

A less involved method of feature retention is simply to keep those above some threshold. Thresholding features for retention is a common theme in many signal processing algorithms. In some applications, thresholding may be rather unprincipled. We, however, carefully set our threshold so that it has physical meaning in relation to our data. Setting the threshold to 0.001 will include all input dimensions with at least an importance of one tenth of one percent (assuming the l_1-norm is used when normalizing the computed relevances). Further, a threshold of 0.001 is considerably smaller than an equal weighting of the input dimensions which would give a relevance of ≈ 0.005 to each dimension of our nearly 200-dimensional data.

4.3 Classification and feature extraction results for GRLVQ(I)

We present the results of GRLVQ(I) on the LCVF data set described in Section 3.2. Classification results are presented in the first part of this section while results of the computed relevances as well as an assessment of feature quality using an independent MED classification are presented in the second part.

4.3.1 Evaluating GRLVQ(I) as a classifier

In this section, we present the classification result of GRLVQ(I) on a series of classification problems of increasing difficulty. We start with the 7-class problem to evaluate the in-class conditional update rule and work our way to the 23-class and 35-class problem to evaluate the combined effects of our in-class conditional update and our adaptation of conscience learning for the supervised classification setting.

The 7-class problem of Section 3.2 is a relatively simple classification problem used to

isolate the effects of our in-class conditional update rule. In evaluating this problem, we do not use conscience learning. The benefit of our in-class conditional update is illustrated in Fig. 4.1 where we see an $\approx$ 35% faster convergence than GRLVQ, to the maximum classification accuracy compared to GRLVQ.

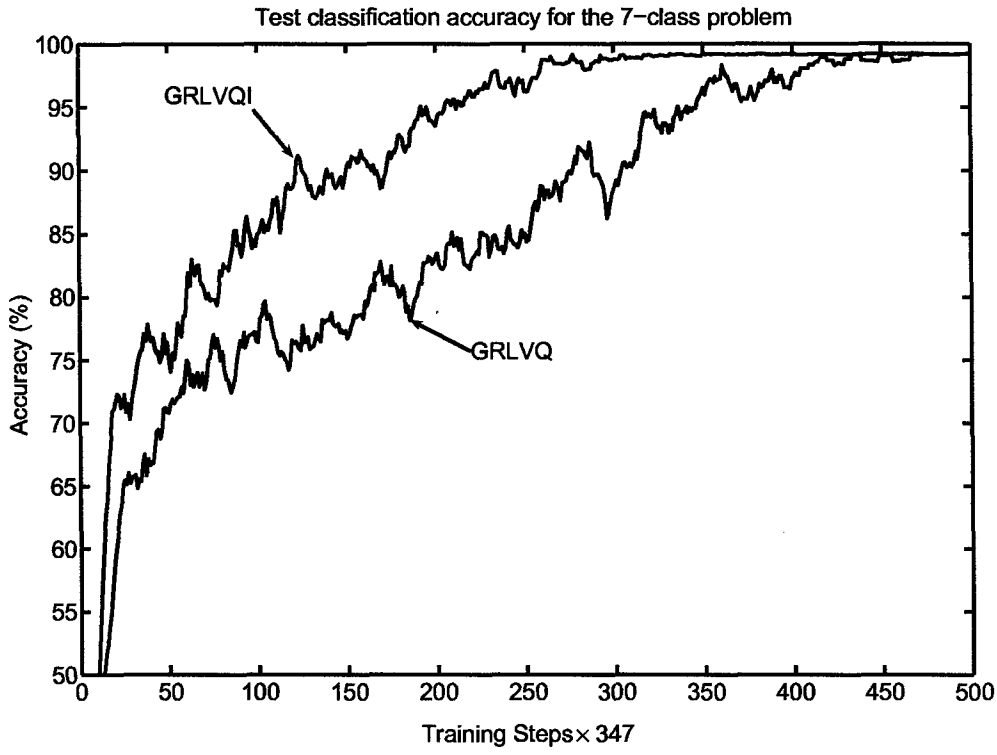


Figure 4.1 : Test accuracy of GRLVQI with the in-class conditional update (without conscience learning) and of GRLVQ for the 7-class data set, on test data. Curves are the average of the three independent jack-knife runs, in each of which the accuracy is computed as the average of the individual class accuracies as described in Section 4.1.

Once we incorporate conscience learning, the speedup benefit from our in-class conditional update is absorbed in the extra processing time required to ensure all prototype vectors learn. As a result, we do not compare the speedup differences between GRLVQI and GRLVQ. Table 4.1 tabulates the results for GRLVQ(I) for the three classification problems evaluated. Adding conscience learning to the in-class winner selection increases classification accuracy from 95.1% to 97.0% for the 23-class problem. For the 35-class problem, we achieve a more impressive 5.6% gain in classification accuracy from 91.6% to 97.2%. In both cases,

the performance is better than that of the hybrid ANN (92.1%) used for a benchmark classification in Section 3.3.

7-Class Problem

	Acc_{run1}	Acc_{run2}	Acc_{run3}	Acc_{mean}	# Features
GRLVQ	99.1%	99.9%	99.9%	99.3%	45
GRLVQI	99.1%	99.7%	99.9%	99.2%	45

23-Class Problem

	Acc_{run1}	Acc_{run2}	Acc_{run3}	Acc_{mean}	# Features
GRLVQ	95.7%	94.9%	94.6%	95.1%	83
GRLVQI	97.7%	95.3%	97.8%	97.0%	81

35-Class Problem

	Acc_{run1}	Acc_{run2}	Acc_{run3}	Acc_{mean}	# Features
GRLVQ	91.8%	90.6%	92.6%	91.6%	109
GRLVQI	98.1%	97.4%	96.0%	97.2%	107

Table 4.1 : Classification accuracy achieved by GRLVQ(I) for each of the three independent jack-knife runs, their average, and the average number of features with relevances ≥ 0.001 for the 7-class, 23-class, and 35-class problems.

We achieve better classification accuracy with GRLVQI over GRLVQ because we make better use of the available classification resources. Improved resource utilization results in better boundary definition which in turn affects the spectral components GRLVQ(I) discovers for classification. Recall from Section 2.3.2.2 that GRLVQ trained 52% (38%) of the prototype vectors for the 23-class (35-class) problem. We verify that adding conscience to the in-class winner selection ensures 100% of the prototype vectors learn for both problems. The number of prototype vectors that learn are the number of prototypes that are different from their initial random state prior to training than their final converged state after training has completed. Even if a prototypes is updated during training, it may not participate in the final solution. That is, not all prototype vectors that learned during training have input samples assigned to them once the classifier has converged.

In Section 3.3 we provided a benchmark classification for each of the three independent jack-knife runs of the 23-class problem. The goal here is not to compare the GRLVQ(I)

classifier to the hybrid ANN, rather compare our results with a challenging benchmark set by proven methods. The hybrid ANN classification accuracy of 92.1% are close to that of both GRLVQ and GRLVQI. It is reasonable to expect GRLVQ(I) would achieve higher classification accuracy than the hybrid ANN achieves on all available (194) spectral features since GRLVQ(I) discovers the most important features for classification and suppresses the superfluous ones.

Based on the classification results, we feel the classification performance of our GRLVQI over GRLVQ for remotely sensed hyperspectral data with a relatively large number of classes is significant. Although simply looking at the overall classification accuracy supports this claim, we recognize it is only fair to make such statements after an in-depth comparison of GRLVQ and GRLVQI results using acceptable techniques. A brass-tacks comparison of GRLVQI and GRLVQ is the topic of Chapter 5.

4.3.2 Evaluating GRLVQ(I) as a feature extractor

The second aspect of evaluating the joint classifier and feature extractor is to analyze the extracted features. Two aspects of feature extraction are important. First is to determine how many features to retain. Second is to determine how good the features are at either preserving or enhancing class discrimination capability.

4.3.2.1 Comparison between GRLVQ and GRLVQI computed relevances

We observe from Table 4.1 that as the number of classes in our classification problem increases, so does the number of spectral components that GRLVQ(I) requires to distinguish those classes. This comes at no surprise since the geometric relationship between neighboring classes is likely more complex. To determine which input dimensions are important for a given classification, one must view the relevance factors computed by GRLVQ(I) for each of the (194) input dimensions.

Relevance factors and mean spectra of representative classes for the 7-class, 23-class, and 35-class problems are plotted in Fig. 4.2, Fig. 4.3, and Fig. 4.4 respectively for GRLVQI only. Spectral features corresponding to large relevance factors for the 7-class problem fall entirely below 0.9μm as is shown in Fig. 4.2. The 23-class problem requires considerably more spectral features than the 7-class problem. Spectral ranges deemed important by GRLVQI for the 23-class problem include 0.44μm to 0.68μm, 0.71μm to 1.11μm, and 1.42μm to 1.68μm (Fig. 4.3). For the 35-class problem, there are four important spectral regions: 0.44μm to 0.68μm, 0.71μm to 1.11μm, 1.42μm to 1.68μm, and 2.1μm to 2.24μm and additional spurious relevances in the 1.94μm to 2.48μm range (Fig. 4.4).

As the classification problem becomes more difficult, GRLVQI must move its relevance resources around to capture the information important for accurate classification of the data. Some general observations can be made based on the placement of GRLVQI-computed relevances for each increasingly difficult classification problem. To facilitate our discussion, we will use A1 to indicate the area above 0.4μm but below 1.35μm, which is the lower boundary of the first data fallout region. The area above 1.4μm and below 1.8μm, which are the upper boundaries of the first data fallout region and lower boundary of the second data fallout region respectively, will be referred to as A2. Similarly, A3 is the area above 1.94μm, which is the upper boundary of the second data fallout region. For the easier 7-class problem, GRLVQI places its relevance resources entirely in A1 where the spectral curves appear most different. In the 23-class problem, GRLVQI expands its emphasis in A1 while adding relevances to A2 where the spectral curves appear least different. In the 35-class problem, GRLVQI further expands its emphasis in A1 and A2 and further places relevances in A3 where the difference in the spectral curves is greater than that of A2 and appears to have similar differences between spectral curves as A1 except the order of the plots are reversed.

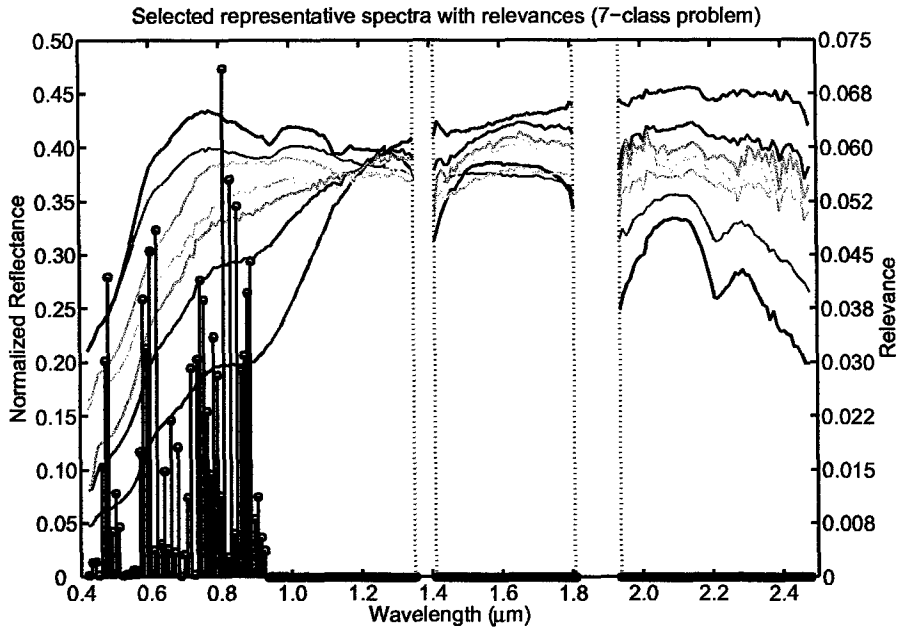


Figure 4.2 : Average spectra for classes A (red), D (blue), E (black), H (orange), I (purple), L (magenta), and W (green). Relevance factors are the averages of three jack-knife runs obtained by GRLVQI (black stem plot) for the 7-class problem. The dotted vertical lines indicate data fallout due to saturation of the atmospheric water bands.

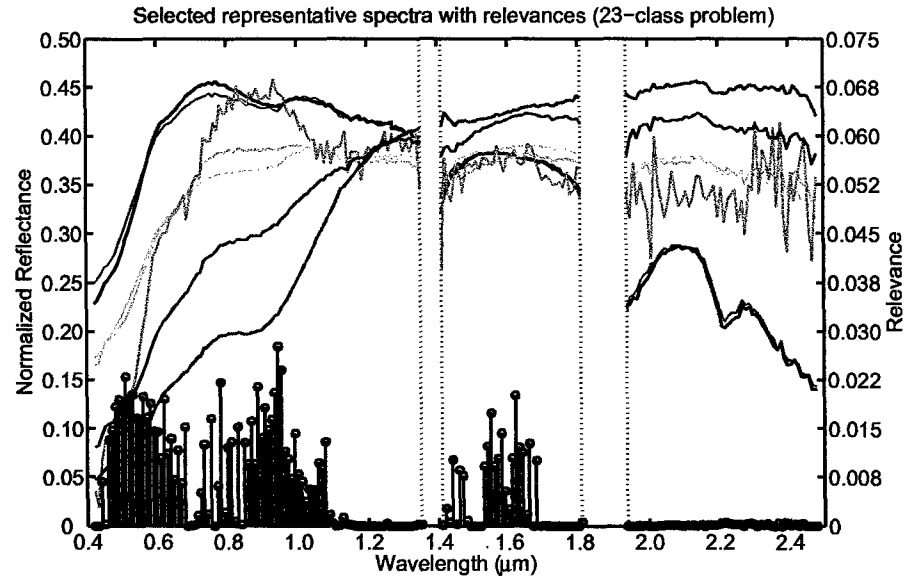


Figure 4.3 : Average representative spectra of classes A (red), G (green), H (orange), L(magenta), O (purple), Q (black), and R (blue). Relevance factors are the averages of three jack-knife runs obtained by GRLVQI (black stem plot) for the 23-class problem. Classes were selected to show largest diversity for this display. The dotted vertical lines indicate data fallout due to saturation of the atmospheric water bands.

We compare relevance factors of GRLVQI (black) to relevance factors of GRLVQ (red) for the 23-class problem in Fig. 4.5. Here, GRLVQ places emphasis in the 1μm to 1.1μm range where GRLVQ does not. Other differences are easier to see by plotting the difference between GRLVQI and GRLVQ relevances as in Fig. 4.7 (top). Additional emphasis by GRLVQI occurs in two regions: 0.55μm to 0.75μm and 1.55μm to 1.6μm. Here we clearly see GRLVQ place greater emphasis in four wavelength regions: 0.45μm to 0.55μm, 0.8μm to 0.95μm, 1.4μm to 1.55μm, and 1.6μm to 1.7μm. Both GRLVQI and GRLVQ place the same emphasis in the 1.1μm to 1.35μm and the 1.7μm to 2.5μm regions. Several general observations can be made regarding the differences in relevances for this 23-class problem. The differences in relevances calculated by GRLVQI and GRLVQ appear in clusters and tend to be fairly significant in A1. In A2, the differences are more spurious but it is clear that GRLVQ place more importance in this area than GRLVQI. There is essential no difference in relevance placement by GRLVQI and GRLVQ in A3.

We similarly compare the relevances for the 35-class problem in Fig. 4.6. We see that GRLVQI places emphasis in the 2.1μm to 2.25μm range where GRLVQ does not. Considerably more emphasis is placed by GRLVQI in the 0.6μm to 0.7μm range and spurious emphasis in the 1.0μm to 1.15μm region. We see that GRLVQ places emphasis in the 0.75μm to 0.85μm range and the 1.4μm to 1.5μm range. Other differences exist and are illustrated more clearly in Fig. 4.7 (bottom). Here we see that GRLVQ has more relevance placed than does GRLVQI in the following regions: 0.55μm to 0.6μm, 0.7μm to 0.85μm, and spurious amounts in the 1.4μm to 1.8μm region. The difference of relevances between GRLVQI and GRLVQ in A1 appear more spurious than in the 23-class problem and the clustering of the differences are not as significant (large). The exception is the region from 0.6μm to 0.7μm where GRLVQ places little of its relevances resources. In A2, we see that GRLVQ consistently places more of its relevance resources than does GRLVQI. The opposite appears to be true in A3. So

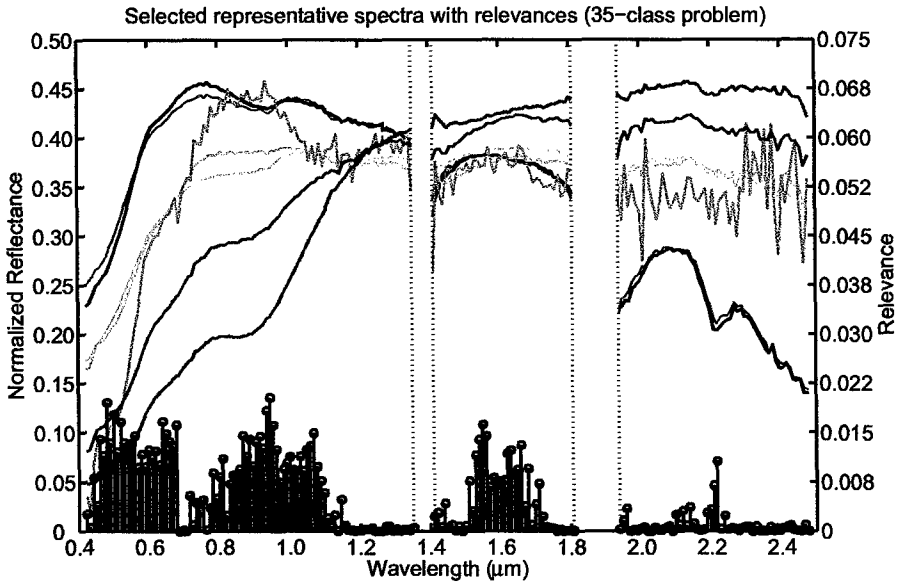


Figure 4.4 : Average representative spectra of classes A (red), G (green), H (orange), L(magenta), O (purple), Q (black), and R (blue). Relevance factors are the averages of three jack-knife runs obtained by GRLVQI (black stem plot) for the 35-class problem. Classes were selected to show largest diversity for this display. The dotted vertical lines indicate data fallout due to saturation of the atmospheric water bands.

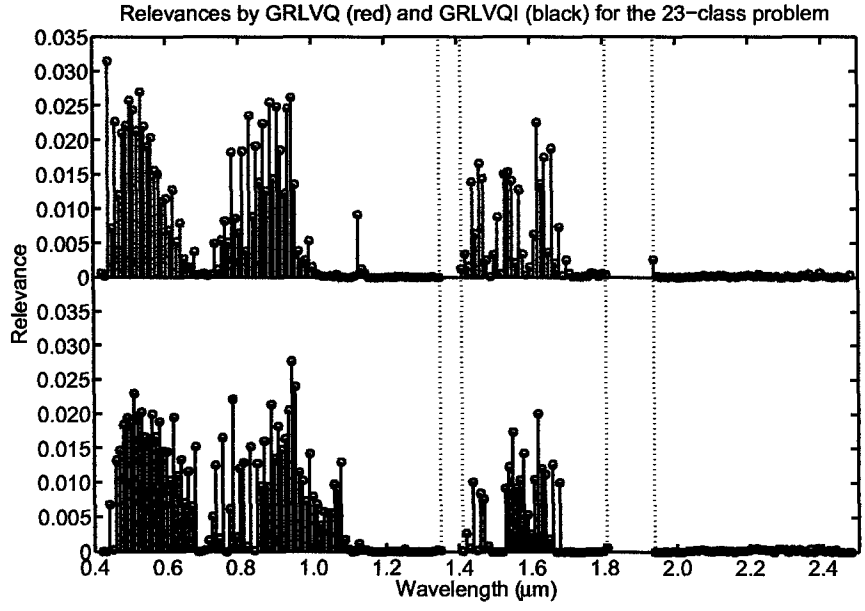


Figure 4.5 : Relevance factors from GRLVQI (black stem plot) with those obtained by GRLVQ (red stem plot) for the 23-class problem. The dotted vertical lines indicate data fallout due to saturation of the water bands.

far, we have seen that GRLVQ(I) places relevance resources at inflection points (e.g., 0.8μm and 0.9μm for Class A (red) in Fig. 4.4) and not at locations related to the overall mean signatures or their standard deviations. In Fig. 4.8, we see that a dip in the overall mean (across all training data) near 2.2μm coincides perfectly with the large relevance placed by GRLVQI near 2.2μm (the first dashed vertical line). The standard deviation peak near 2.2μm is occurs one band later (the second dashed vertical line near 2.2μm).

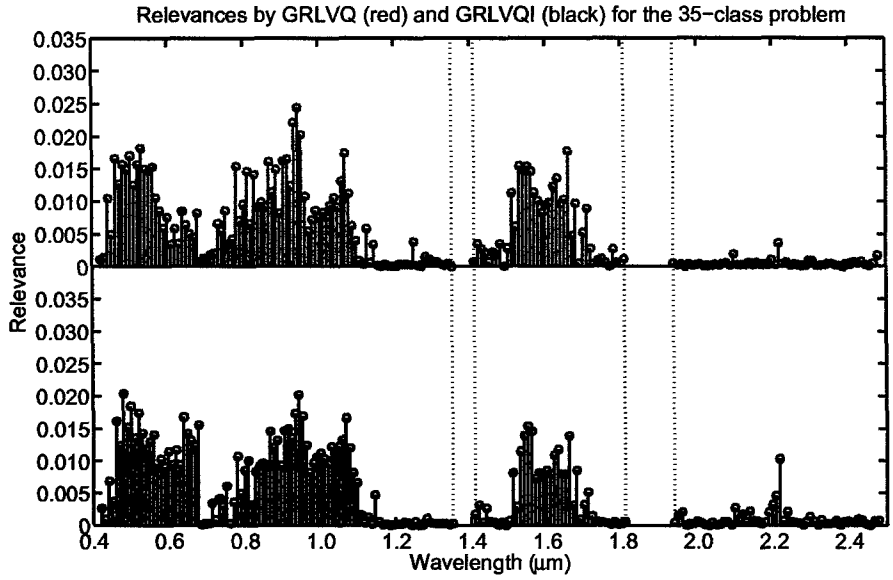


Figure 4.6 : Relevance factors from GRLVQI (black stem plot) and those obtained by GRLVQ (red stem plot) for the 35-class problem. The dotted vertical lines indicate data fallout.

4.3.2.2 GRLVQ(I) feature retention and discrimination capability with the MED classifier

In Table 4.2, we list the maximum MED classification accuracy using relevance selected features as described in Section 4.2. We provide a baseline MED classification using all available (original) 194 spectral features. We clearly see from these results that features discovered by GRLVQ(I) are indeed a better set of features for classification than all available (194) spectral features. This evaluation is only for the 23-class and 35-class problems as these data sets were used to evaluate the difference between GRLVQ and GRLVQI where the 7-

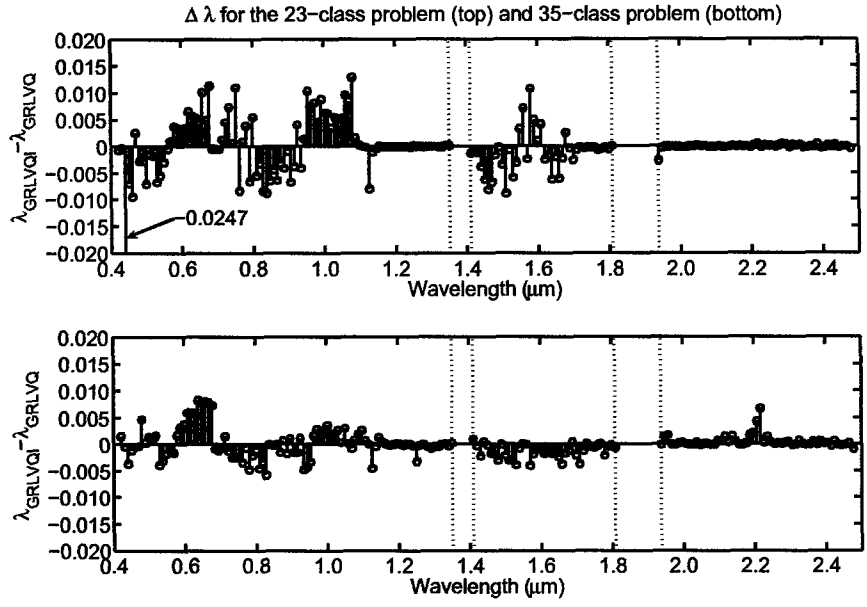


Figure 4.7 : **Top:** Plot of $\lambda_{GRLVQI} - \lambda_{GRLVQ}$ for the 23-class problem. **Bottom:** Plot of $\lambda_{GRLVQI} - \lambda_{GRLVQ}$ for the 35-class problem. The dotted vertical lines indicate data fallout due to saturation of the water bands.

class was to demonstrate the speed-up potential of the in-class conditional update rule.

Relevance selected features used in the MED classifier

	23-Class problem		35-Class problem	
	Accuracy	#Features	Accuracy	#Features
GRLVQI Features	96.4%	52	91.8%	191
GRLVQ Features	96.2%	45	91.7%	184
Baseline MED	92.8%	194	91.1%	194

Table 4.2 : An independent verification of GRLVQ(I) feature quality using the Minimum Euclidean Distance (MED) classifier. Classification accuracy is averaged over independent jack-knife runs. The accuracy in each independent jack-knife run is calculated as the mean of the individual class accuracies.

The peak performance and corresponding number of features presented in Table 4.2 is one view of the MED results. In Fig. 4.9, we show MED classification accuracy versus the number of retained features in descending order of relevance. For the 23-class problem, the MED classifier is able to make significant improvements in classification accuracy using relatively few GRLVQ(I)-based relevance selected features. Improvements for the 35-class problem using GRLVQ(I)-based relevance selected features is not as dramatic. This is not an

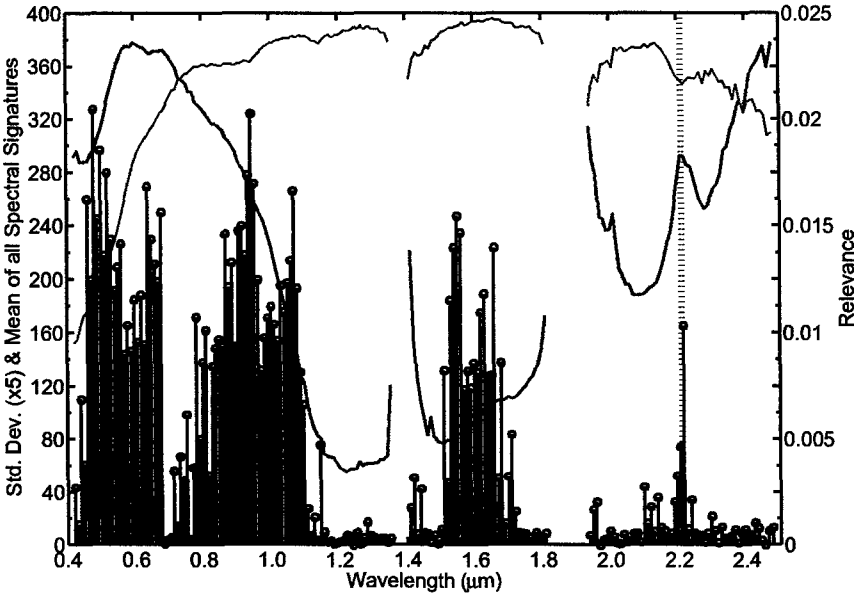


Figure 4.8 : Comparison of the overall mean (blue) and standard deviation (red) of the training set and the GRLVQI computed relevance (black stem) for the 35-class problem. The first dashed line indicates the dip in the mean which coincides perfectly with the peak in the GRLVQI computed relevance. The second dashed line indicates the peak in the standard deviation which occurs one band later. Other relevances appear to follow inflection points in the mean spectral and/or the standard deviation of the spectra.

indication that the GRLVQ(I) features are poor, it is a limitation of the MED in evaluating the GRLVQ(I) features as discussed in Section 4.2.

4.4 Summary and Discussion

From the original 194 data dimensions, GRLVQ(I) discovers a significantly reduced set of features relevant for classification. The feature set is clearly meaningful for GRLVQ(I) as they both achieve very high classification accuracies. We see that as the classification problem becomes more complex, GRLVQ(I) requires additional features to distinguish between increasing number of classes. We surmise that the highly correlated spectral bands limit the reduction of features GRLVQ(I) requires for classification. This is seen from the spectral plots with relevance factors in Fig. 4.2, Fig. 4.3, and Fig. 4.4. We further support this claim based on the observation that relevances next to large (small) relevance values are also large

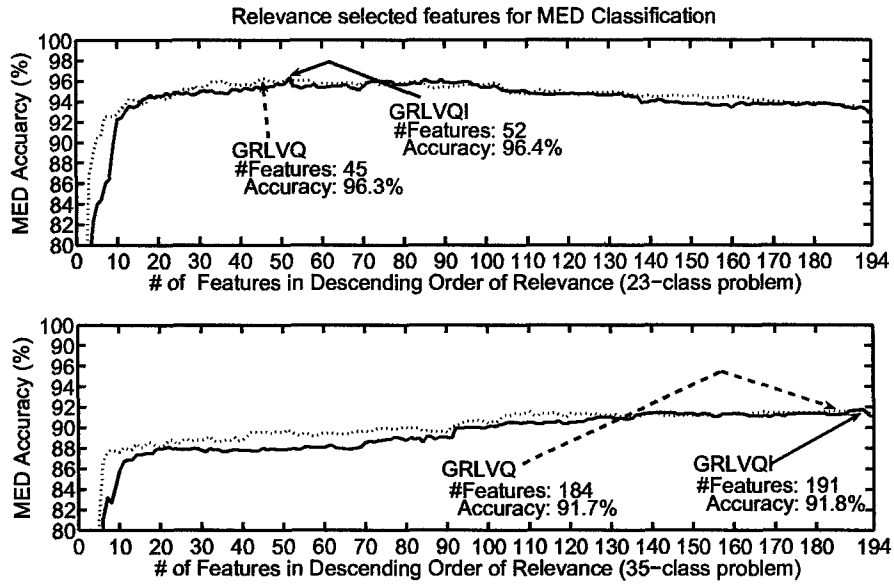


Figure 4.9 : **Top:** Plot of MED classification accuracy versus relevance selected features for GRLVQ computed relevances (dashed) and GRLVQI computed relevances (solid) for the 23-class problem. **Bottom:** Plot of MED classification accuracy versus relevance selected features for GRLVQ computed relevances (dashed) and GRLVQI computed relevances (solid) for the 35-class problem.

(small).

Deciding how many features to retain is an age-old problem. Many signal processing tools rely on the N "most important" features. For the Fourier series and Principal Component Analysis, these are the first N coefficients; for wavelet analysis, these are the N largest magnitude wavelet coefficients. In GRLVQ(I), we keep features based on the N largest relevance factors. The thresholding method is a very rudimentary technique (yet commonly used in many signal processing methods for feature retention) which does not take into account the trade-off between the classification accuracy and the number of features. A better method was to use the MED classifier to help us choose how many features to retain. This aspect of relevance learning (as with many other signal processing methods) is an open area for future research.

The MED assessment provided us with a first-order confirmation of the class discrimination capability of the GRLVQI feature set. A classifier more similar to the GRLVQ(I)

would be a better method for the independent evaluation of GRLVQ(I)-selected features. For example, a K-means classifier where the value number of means K is equal that of the number of GRLVQI prototype vectors used per class. One may use the K prototypes from GRLVQ(I) as the initialization point for each of the K-means and use the relevance weighted Euclidean distance for the distortion measure (instead of the unweighted Euclidean distance). The independent assessment of the GRLVQ(I) features is an open area for future research.

The MED classifier is a relatively simple classifier which benefited greatly from the feature extraction capabilities of GRLVQ(I). Other classifiers could also benefit from the extracted feature set and may yield improved classification results. For example, the Maximum Likelihood classifier may be unusable for high-dimensional data because it lacks a sufficient number of training samples, which number is dependent on the number of input features used (see a case study in Merényi [22]). However, ML could become applicable after GRLVQ(I) feature extraction with better results than in [22]. We used the MED as independent evaluator because of its cost effectiveness and because it does not require guesses for additional parameters (such as prior probabilities, etc). We surmise that more sophisticated classifiers could benefit similarly, and perhaps produce better classification than that of GRLVQ(I).

In the next chapter, we compare the achievements of GRLVQ and GRLVQI on the classification results from the present chapter. This is accomplished by providing both a micro-comparison of individual classes and a macro-comparison of overall classifier performance. In the end, we show that, although GRLVQ performs well for hyperspectral data, our GRLVQI is superior for complex data sets with hundreds of input dimensions.

Chapter 5

Performance improvement of GRLVQI over GRLVQ

We hypothesize that the performance of GRLVQI is significantly better than that of GRLVQ for classifying hyperspectral data. In order to show that our hypothesis is correct, we consider several methods of evaluating the improved classification results. First, we consider theory on generalization bounds to compare the upper bounds on the generalization error of GRLVQ(I) to serve as an indicator of the expected performance of each classifier. Second, we use *commission errors* and *omission errors* for a class-by-class performance comparison between GRLVQI and GRLVQ. Third, we use the widely accepted κ statistic to remove that part of the classification achieved due to pure chance, giving a "normalized" measure of agreement between each classifier and the labeled test data. The omission and commission errors as well as the κ statistic are commonly used in the remote sensing community to evaluate a classifiers performance against known data. The fourth method of comparison is the *Wilcoxon Signed Ranks Test* (WSRT). The WSRT allows us to directly measure the statistical significance of GRLVQI's improvement over GRLVQ.

Although the main focus of this chapter is to quantify the classification improvement of GRLVQI over GRLVQ, the classification results are affected by the learned weighting of the input dimensions. Consequently, the analysis accomplished in this chapter implicitly also evaluates the quality of the feature set for classification. See Section 4.3 for details on the relevances learned by GRLVQ(I) for the Lunar Crater Volcanic Field (LCVF) data set evaluated in this thesis.

5.1 Background – classifier performance evaluation methods

5.1.1 Using generalization bounds to compare classifier performance

Recent publications offer results on generalization bounds specifically for LVQ-type classifiers. Crammer et al. derive an upper bound for the generalization error for the LVQ2.1 family of classifiers. The generalization error is the empirical error over the training samples plus a term that is a function of the Vapnik- Chervonenkis (VC) dimension [14], and is dependent on the dimensionality of the data. The empirical error is based on the maximal margin principle [14] and penalizes margins which are smaller than some threshold θ. However, Hammer et al. [47] state that the VC-dimension approach is not valid for LVQ classifiers with an adaptive diagonal metric such as RLVQ [59] and GRLVQ, and hence GRLVQI.

Using the Radamacher-Gaussian complexity described in Bartlett and Mendelson [46], Hammer et al. [47] derive upper bounds on the generalization error of LVQ classifiers that use the winner-takes-all rule and include classifiers with an adaptive diagonal metric. This formulation is independent of the dimensionality of the data. The derived generalization error is the sum of three terms. The first term is a function of the empirical error over the training samples, which penalizes misclassifications and small margins. The second term is an empirical Gaussian complexity term which can be influenced by the magnitude of the training samples $\mathbf{x}^m$ or the converged weights, whichever has the largest l_2 norm. The third term depends on the number of training samples and the confidence in the estimate. These last two terms are inversely proportional to the margin and hence favor large margins.

Unfortunately, the current theory assumes a two-class problem and is invalid for the multi-class case. However it may be possible to compare the empirical error of each classifier in order to gain hints as to which classifier will generalize better [B. Hammer, private communication]. If the difference in the empirical error is large, then the generalization bounds

can serve as an indicator that the classifier with the lower error is expected to perform better than the classifier with the larger error. In Section 5.2.1, we will use a definition of the empirical error that reflects the margin maximized by GRLVQ(I) to show that GRLVQI is expected to generalize better than GRLVQ.

5.1.2 The confusion matrix and its error measures

In this section we present an example that illustrates the construction of a confusion matrix explain how to use the confusion matrix to determine the commission and omission errors which are used in Section 5.2.2. We further label important row and column values used to calculate the κ statistic in Section 5.1.3.

An entire picture of a classifiers performance can be viewed using a confusion matrix (see [23,60] for additional details). The confusion matrix provides five quantities: commission error (CE), omission error (OE), producer's accuracy (PA), consumer's accuracy (CA), for each class, and overall classification accuracy. These are defined as:

- **CE:** The percentage of samples classified as a given class but not belonging to that class.
- **OE:** The percentage of samples belonging to the given class but omitted (misclassified to other classes).
- **PA:** The accuracy of the prediction from the perspective of the classifier. It gives the percent of the total number of samples that belong to a given class that are actually classified as that class.
- **CA:** The accuracy of the prediction based on the user's perspective. It gives the percent of what is labeled as belonging to a given class is actually a member of that class.

Consider a hypothetical 3-class problem where *Class A* has 17 samples, *Class B* has 13 samples, and *Class C* has 18 samples. After training our classifier on our hypothetical

problem, we report the classification in a confusion matrix (Fig. 5.1).

Confusion Matrix for Hypothetical 3-Class Problem

	Classified As A	B	C	#Samples $x_{col(s)}$	PA	OE
A	12	3	2	17	$\frac{12}{17} = 70.6\%$	$\frac{3+2}{17} = 29.4\%$
B	2	11	0	13	$\frac{11}{13} = 84.6\%$	$\frac{2+0}{13} = 15.4\%$
C	1	0	17	18	$\frac{17}{18} = 94.4\%$	$\frac{1+0}{18} = 5.6\%$
Totals $x_{row(s)}$	15	14	19	Diag. Sum $\sum_{s=1}^{3} x_{s,s} = 40$		
CA	12/15 =80.0%	11/14 =78.6%	17/19 =89.5%	NEWA= $\frac{40}{17+13+18=48} = 83.3\%$		
CE	$(2+1)/17$ =17.6%	$(3+0)/13$ =23.1%	$(2+0)/18$ =11.1%	EWA= $\frac{12/17+11/13+17/18}{3} = 83.2\%$		

Figure 5.1 : Confusion matrix of a hypothetical three class problem. The matrix presents the Commission Error (CE), Omission Error (OE), Consumer's Accuracy (CA), Producer's Accuracy (PA), for each class and both the *equal weighted* and *non-equal weighted* classification accuracies (EWA and NEWA, respectively) described in Section 4.1. The values $x_{s,s}$, $x_{row(s)}$, and $x_{col(s)}$ are used to label the diagonal and the row and column containing the totals used in Section 5.1.3 on the κ statistic.

Each row in the confusion matrix is the classification for the class label heading that row. For *Class A*, 12 samples were correctly classified as *Class A* and 5 samples were incorrectly classified, three of which are assigned to *Class B* and 2 to *Class C*. For *Class A*, 12 of 17 samples were correctly classified, therefore the PA $= \frac{12}{17} = 70.6\%$. The OE for *Class A* is the number of misclassified samples divided by the total number of samples. For *Class A*, 3 samples were classified as *Class B* and 2 as *Class C* which gives an OE $= \frac{3+2}{17} = 19.4\%$. Note that PA+CA=100% and that the PA and OE are based solely on row information.

Each column in the confusion matrix is the number of samples classified as the label heading that column. For *Class A*, 12 samples were correctly classified as *Class A* and 3 samples from other classes were misclassified as *Class A* (2 from *Class B* and 1 from *Class C*). These values are used to find the CA and the CE. For the CA, we find the ratio between *Class A* samples classified correctly to the total number of samples classified as *Class A*, which gives a CA $= \frac{12}{15} = 80.0\%$. Here, the CA is based entirely on column information. The

CE is a mixture of row and column information and is the ratio of samples commissioned into the class to the number of samples in the class. For *Class A* we have 17 *Class A* samples plus 2 samples commissioned from *Class B* and 1 from *Class C*. This yields a CE $= \frac{2+1}{17} = 17.6\%$. Note that the sum of the column statistics are not 100% unlike the row statistics.

5.1.3 κ statistics

Omission and commission errors on a class-by-class basis illustrate where the classifier performs well and where it has problems. Some may feel that classification accuracy alone adequately tells the story of classifier performance. However, relying on classification accuracy alone disregards the (potentially high) classification accuracy achieved due to chance [23]. Cohen [61] developed a method to compare the agreement between the outcome of two "judges" (classification results versus truth) on a series of events (samples with known class labels). This method attempts to remove that part of the evaluation attributed to chance agreement giving a more precise measure of the agreement between the prediction of the classifier and the known class labels.

The idea is to compare the *observed correct* to that of the *expected correct* (also called *chance agreement*)

$$\kappa = \frac{\sum_{s=1}^{S} p_{s,s} - \sum_{s=1}^{S} p_{row(s)} p_{col(s)}}{1 - \sum_{s=1}^{S} p_{row(s)} p_{col(s)}}, \tag{5.1}$$

where $s \in \{1, ..., S\}$ and S is the number of classes, $p_{s,s}$ is the agreement between the known and the predicted, and $p_{row(s)}p_{col(s)}$ are the marginal probabilities. In the case where the number of samples M is fixed (and according to Bishop et al. [62], a multinomial sampling model, which is the case in a multi-class classification problem under the assumption that each class is described by a different distribution), one can obtain a maximum likelihood

estimate of the true κ by using the observed values directly from the confusion matrix:

$$\hat{p}_{s,s} = \frac{x_{s,s}}{M}, \tag{5.2}$$

$$\hat{p}_{row(s)} = \frac{x_{row(s)}}{M}, \tag{5.3}$$

$$\hat{p}_{col(s)} = \frac{x_{col(s)}}{M}. \tag{5.4}$$

Substituting the values of $\hat{p}$ from above into Eq. 5.1 gives us an estimate of the κ statistic, $\hat{\kappa}$ [60,62]

$$\hat{\kappa} = \frac{\sum_{s=1}^{S} \frac{x_{s,s}}{M} - \sum_{s=1}^{S} \frac{x_{row(s)} x_{col(s)}}{M^2}}{1 - \sum_{s=1}^{S} \frac{x_{row(s)} x_{col(s)}}{M^2}}. \tag{5.5}$$

Capturing the essence of Eq. 5.5 in words [23,60,62]

$$\frac{\text{Actual Agreement} - \text{Chance Agreement}}{1 - \text{Chance Agreement}} \tag{5.6}$$

The *Actual Agreement* is as it sounds, and is simply the achieved classification accuracy (term $\sum_{s=1}^{S} \frac{x_{s,s}}{M}$ of Eq. 5.5). The *Expected Correct* is an estimate of chance agreement contributing to the *Observed Correct* [23] (term $\sum_{s=1}^{S} \frac{x_{row(s)} x_{col(s)}}{M^2}$, which are the marginal distributions). The terms $x_{row(s)}$ and $x_{col(s)}$ are the actual counts appearing in the row and column marked "Total" and "# Samples", respectively, in Fig. 5.1. The term $x_{s,s}$ is the s^{th} diagonal entry while the value of M is the total number of samples evaluated.

We use the κ statistic to compare the relative performance between two classifiers measured against pure chance. That is, we interpret the observation of GRLVQI having a larger κ statistic than GRLVQ as having more confidence in GRLVQI's results over the results obtained by GRLVQ. This is a reasonable approach since the κ statistic is normalized, removing that part of the classification resulting from pure chance [23, 60–62]. We use the estimate $\hat{\kappa}$ of Eq. 5.5 to estimate the true κ statistic for the 23-class and 35-class problems in Section 5.2.3.

5.1.4 Wilcoxon Signed Ranks Test

The κ statistic provides us with one statistical method of evaluating classification performance. It does not, however, provide a comparison between two classifiers. There is a vast literature for the comparison of two algorithms on (many) different data sets (see, e.g., Demšar [50] and Salzberg [63] and references therein). One common method for comparing two classifiers on multiple data sets is the *paired t-test.* Demšar [50] notes that the *paired t-test* is ill-suited for the task for several reasons. First, the outcome of the test only makes sense if the difference between the classifiers is substantial (although it is not defined what constitutes substantial). Second, it assumes that for relatively few classification test results ($\approx$30 according to [50]), that the classification accuracy results are normally distributed. Since one generally has fewer than 30 sets of classification results, and common methods for determining the normality of small sample sizes are not reliable, we do not consider the paired t-test. Furthermore, the paired t-test is sensitive to outliers.

There are several tests which make no assumptions on the distribution of the classification results, one of which is the binomial test described by Salzberg in [63]. The binomial test, however, is a relatively week test [50, 63], as it does not take into account the agreement between the algorithms (see Section 5.1.3 on Cohen's κ statistic as a measure of agreement) nor does it take into account the quantitative differences between algorithm results. A second method is the *Sign Test*, which is considered a weaker test than the *Wilcoxon Signed Ranks Test* [50] because it does not weigh the magnitude of the difference in the results, it only acknowledges that a difference occurred.

The *Wilcoxon Signed Ranks Test* (WSRT) [50, 64] (for paired tests) is used in this thesis to test the statistical significance of GRLVQI's improved classification performance over GRLVQ. The WSRT ranks and weighs the difference in classification results between GRLVQI

and GRLVQ. It places greater emphasis on larger differences than on smaller ones making no assumptions on the distribution of the resulting classification accuracies, and suppresses the negative effects of outliers. The WSRT defines three quantities. The first quantity is the number of trials N_t. The second and third quantities are the sums of the rankings assigned to the absolute differences in classification accuracies. The positive sum of ranks ($W+$) is the sum of the ranks assigned to positive differences while the negative sum of ranks ($W-$) is the sum of the ranks assigned to negative differences. For example, if the classification accuracies of four tests were {70%, 95%, 95%, 88%} for GRLVQI and {90%, 85%, 65%, 98%} for GRLVQ, then the difference of their outcomes would be {−20%, 10%, 30%, −10%}. In the WSRT, one would rank the magnitude of the results giving {3, 2, 4, 1} and count the positive ranks as $W+$ and the negative ranks as $W-$, splitting the tied ranks. This would give $W+ = 4 + (2+1)/2 = 5.5$ and $W- = 3 + (2+1)/2 = 4.5$. The number of tests in this example is $N_t = 4$. Should there be a differences of zero (i.e., equal performance), the zero ranks are split evenly amongst $W+$ and $W-$. If there are an odd number of zero differences, one is simply ignored. The statistical significance of the performance difference is then based on the number of tests performed (N_t) and the smallest rank (i.e., $\min(W+, W-)$), which can be found in a table of critical values [65,66] (see Table 5.1 for a portion of the table of critical values from McCornack [66]).

5.2 Results of the performance comparison between GRLVQ and GRLVQI

In this section we use the methodology described in Section 5.1 to evaluate the significance of the improved classification accuracy of GRLVQI reported in Section 4.3 for the LCVF data set discussed in Section 3.2. The commission and omission errors, although not directly discussed in Section 4.3, are derived from analyzing the averaged results of the three independent jack-knife runs for both the 23-class and 35-class problems. The κ statistic

Table of Critical Values of the cumulative one-tail probability
$P[T \leq W + |N_t]$ for the Wilcoxon Signed Ranks Test

		values for α									
N_t	Max	0.075	0.050	0.025	0.020	0.015	0.010	0.005	0.0025	0.0005	0.00005
4	10	0									
5	15	1	0								
6	21	2	2	0	0						
7	28	4	3	2	1	0	0				
8	36	7	5	3	3	2	1	0			
9	45	9	8	5	5	4	3	1	0		
10	55	12	10	8	7	6	5	3	1		
11	66	16	13	10	9	8	7	5	3	0	
12	78	19	17	13	12	11	9	7	5	1	
13	91	24	21	17	16	14	12	9	7	2	
14	105	28	25	21	19	18	15	12	9	4	
15	120	33	30	25	23	21	19	15	12	6	0
16	136	39	35	29	28	26	23	19	15	8	2

Table 5.1 : This table provides a list of critical values for the Wilcoxon Signed Ranks Test for paired tests. It lists the probability that the sum of positive ranks ($W+$) is less than the test statistic (T) given a number of observations (N_t) (i.e., $P[T \leq W + |N_t]$). This table is an excerpt from a much more complete table found in [66].

results discussed below in Section 5.2.3 are derived using the diagonal entries (individual classification accuracies) and the row and column totals of the confusion matrix (discussed in Section 5.1.2 and illustrated in Fig. 5.1 for both the 23-class and 35-class problems. Finally, in Section 5.2.4, the accuracy results are used directly from each of the six independent jack-knife runs from Table 4.1. We have a total of six classification results (three from the 23-class problem and three from the 35-class problem) for both GRLVQI and GRLVQ for use in the Wilcoxon Signed Ranks Test. We feel that the variety of the classifications used in this analysis are adequate to show that the hypothesis that GRLVQI performs better than GRLVQ for the classification of hyperspectral data is correct.

5.2.1 Using margin analysis to infer expected generalization

As stated in Section 5.1.1, we are unable to directly calculate the upper bounds on the generalization error for GRLVQ(I) because these bounds are only valid for the two class

problem. However, we can compare the empirical cost based solely on the hypothesis margins. If the difference in empirical cost between GRLVQI and GRLVQ is large enough, then this provides us an indication that the classifier with the smaller cost is expected to generalize better than classifier with the larger cost.

Crammer et al. show the generalization error is related to an empirical cost function (a measure of the achieved hypothesis margin for the training samples) plus a term that is a function of the Vapnik-Chervonenkis (VC) dimension. Ignoring the VC dimension term (it is the same for both GRLVQ and GRLVQI), we can compare the generalization error by considering only the empirical cost:

$$\mathrm{E}(\{W\}, S)_{CGNT} = \frac{1}{M}|\{m : \Psi_{CGNT}(\{W\}, (\mathbf{x}^m)) < \theta\}| \tag{5.7}$$

where $0 < \theta < \frac{1}{2}$, the margin $\Psi_{CGNT} = \frac{d^K - d^J}{2}$ (d^J and d^K being the squared Euclidean distance between the input sample $\mathbf{x}^m$ and the winning in-class and out-of-class prototypes $\mathbf{w}^J$ and $\mathbf{w}^K$, respectively), S is the training sample set which includes the samples and their respective class labels, and $CGNT$ is a reference to the authors Crammer, Gilad-Bachrach, Navot, and Tishby [14].

Hammer et al. define their cost function somewhat differently:

$$\mathrm{E}(\{W\}, S)_{HSV} = \frac{1}{M}\sum_{m=1}^{M}\begin{cases} 1 & \text{if } \Psi_{HSV} < 0 \\ 1 - \frac{\Psi_{HSV}}{\theta} & \text{if } 0 < \Psi_{HSV} \leq \theta \\ 0 & \text{Otherwise} \end{cases} \tag{5.8}$$

where θ is a constant (here, there is no restriction on θ), the margin $\Psi_{HSV} = d^K - d^J$, and HSV is a reference to Hammer, Strickert, and Villmann.

Based on the two definitions of the margin and the definitions of the cost functions, we see from Fig. 5.2.A and Fig. 5.2.B that GRLVQ is expected to have better generalization. If one defines generalization as the ability to predict the correct outcome of unseen instances,

then this prediction is incorrect as our improved GRLVQI shows better generalization for both the 23-class and 35-class problems.

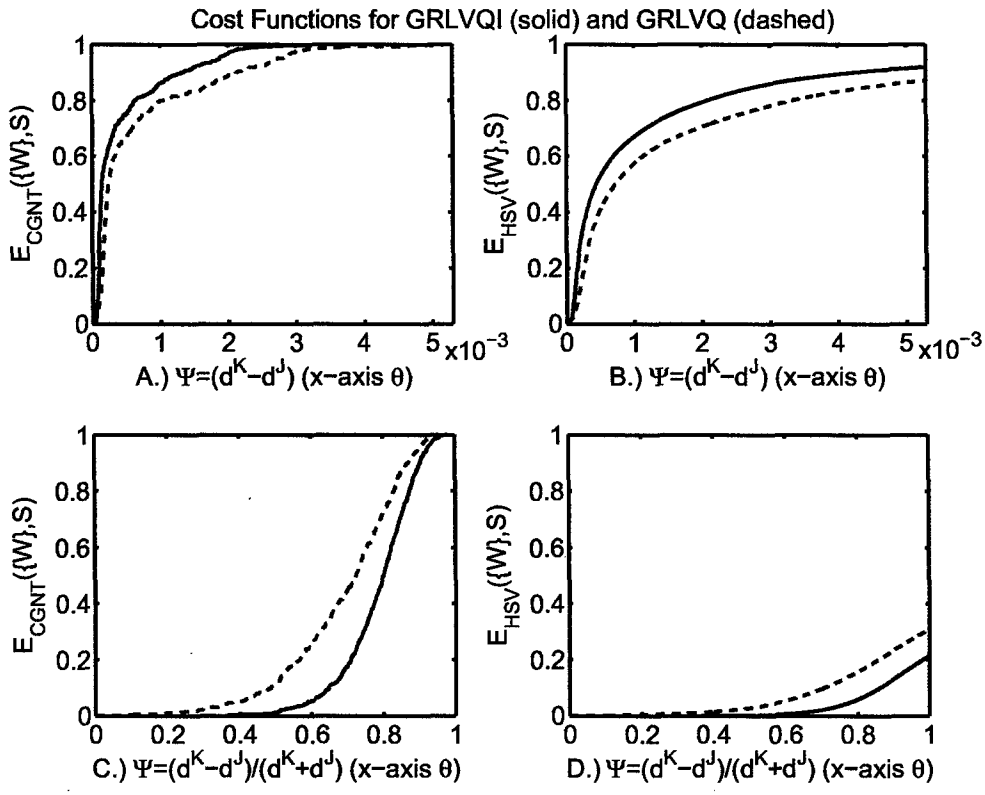


Figure 5.2 : The error, E, is a measure of the sample margin Ψ. Large sample margin results in an E that is small. Small sample margins yield larger E than large sample margins.

This observation can indicate one of three consequences. One is that GRLVQ *really* does generalize better which may not be confirmed until a vast number of unseen samples have been visited. Another is that we cannot assume that the generalization error can be accurately compared using a cost function based on the empirical margin alone. The last is that the cost function used to determine the empirical margin for GRLVQ(I) should more closely follow the work motivated by Crammer et al.

Crammer et al. use their definition of the margin (Ψ) in a cost function which they minimize via gradient descent. That is, they directly maximize their margin, Ψ_{CGN}. Hammer et al. similarly minimize a cost function (Eq. 2.3) which is a function of the misclassifica-

tion measure of Eq. 2.2. The margin (Ψ_{HSV}) appears in the numerator of the misclassification measure. Following more closely the ideas of Crammer et al., it perhaps is more meaningful to define the margin as the misclassification measure of Eq.2.2 (i.e., we redefine the margin $\Psi = \frac{d^K - d^J}{d^K + d^J}$). This definition for the margin (Ψ) gives a direct relationship between the margin and the cost function minimized by GRLVQ(I) more closely matching the ideas of Crammer et al.

What indication of generalization does our definition of margin (Ψ) give? According to Fig. 5.2.C and Fig 5.2.D, GRLVQI has a larger margin than GRLVQ, and one would anticipate a better generalization. We recognize that the difference in margins may not be large enough to make concrete statements regarding the performance difference of GRLVQI and GRLVQ. However, a more relevant definition of the margin as the misclassification measure of Eq. 2.2 indicates GRLVQI would generalize better, which supports the hypothesis that the performance of GRLVQI is significantly better than that of GRLVQ for classifying hyperspectral data.

5.2.2 Errors of omission and commission

It is common to report the quality of a classifier's performance using a single quantitative value that is the classification accuracy. What is missing, however, is how misclassified samples are distributed amongst the remaining classes. We compare the omission and commission errors for the 23-class (Table 5.2 left) and 35-class (Table 5.2 right) problems described in Section 3.2. In our analysis, we do not use the Producers Accuracy or the Consumers Accuracy directly.

From Table 5.2, we can make several observations. First, GRLVQI overall has fewer (significant) omission and commission errors. This observation is portrayed graphically in Fig. 5.3, which shows the percentage of classes along the y axis that have either a commission

	23-Class Problem				35-Class Problem			
	CE (%)		OE (%)		CE (%)		OE (%)	
Class	A	B	A	B	A	B	A	B
A	0.00	0.00	0.00	4.17	0.00	0.00	11.25	3.11
B	9.72	4.17	16.39	9.80	19.52	0.00	22.38	0.00
C	13.07	6.40	14.22	0.52	5.56	2.78	24.44	0.00
D	0.52	0.52	0.52	0.64	4.22	0.00	0.53	1.06
E	0.79	0.00	4.69	0.00	0.00	0.85	11.89	0.93
F	0.00	0.00	4.76	16.67	0.00	0.00	50.13	30.18
G	0.00	0.00	0.00	1.96	0.00	0.00	0.00	0.00
H	0.00	0.00	0.00	6.06	2.08	0.00	5.13	0.00
I	4.17	4.17	11.36	8.33	3.33	0.00	12.47	0.00
J	8.33	8.33	0.00	7.41	0.00	0.00	0.00	6.67
K	35.56	7.04	0.00	2.90	39.91	3.33	3.33	0.00
L	3.55	0.00	7.89	2.47	6.08	3.04	20.52	3.46
M	0.00	8.33	3.70	0.00	0.00	0.00	8.33	8.33
N	0.00	0.00	0.00	0.00	28.33	0.00	13.89	0.00
O	0.00	0.00	0.00	0.00	58.72	4.62	6.67	0.00
P	0.00	3.33	25.00	0.00	7.41	0.00	14.63	0.00
Q	43.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S	0.00	0.00	0.00	20.00	0.00	0.00	0.00	0.00
T	0.00	20.00	6.67	0.00	0.00	0.00	0.00	0.00
U	25.40	19.44	4.17	2.08	25.74	0.00	5.25	7.47
V	0.00	11.11	0.00	0.00	0.00	0.00	4.76	19.44
W	9.29	6.73	2.08	0.00	4.17	4.17	7.49	0.00
X					35.98	27.65	0.00	0.00
Y					29.72	6.39	8.61	4.31
Z					31.03	1.96	0.00	3.33
a					13.33	3.33	3.33	3.33
b					22.04	4.63	1.85	0.00
c					0.00	0.00	50.87	0.00
d					6.95	1.96	16.35	0.00
e					13.54	2.22	0.00	0.00
g					0.00	0.00	0.00	0.00
h					7.33	4.94	2.78	0.00
i					2.22	0.00	3.03	0.00
j					2.22	2.22	0.00	4.08
mean	6.68	4.33	4.41	3.61	10.56	2.12	8.86	2.73

Table 5.2 : Errors of commission and omission for the 23-class and 35-class problems. Columns marked with **A** are GRLVQ results and those marked with **B** are GRLVQI results.

error (left) or omission error (right) greater than some fixed threshold error (E) along the x axis. The results for the 23-class problem (top) shows that none of the classes have more

than 20% commission error for GRLVQI (dotted) where GRLVQ (solid) has commission errors as high as $\approx 45\%$. Omission errors (right) for GRLVQI (dotted) and GRLVQ (solid) show similar performance. This is also indicated in the mean omission error reported in Table 5.2. For the 35-class problem, GRLVQI has far fewer commission and omission errors than GRLVQ (Fig. 5.3 bottom). The significant commission errors (left) rapidly decay for GRLVQI (dotted) and the rate of decay is much slower for GRLVQ (solid). The story is similar for omission errors (right) where GRLVQI (dotted) shows far fewer omission errors than GRLVQ (solid). This is further supported by the mean omission and commission errors for the 35-class problem reported in Table 5.2.

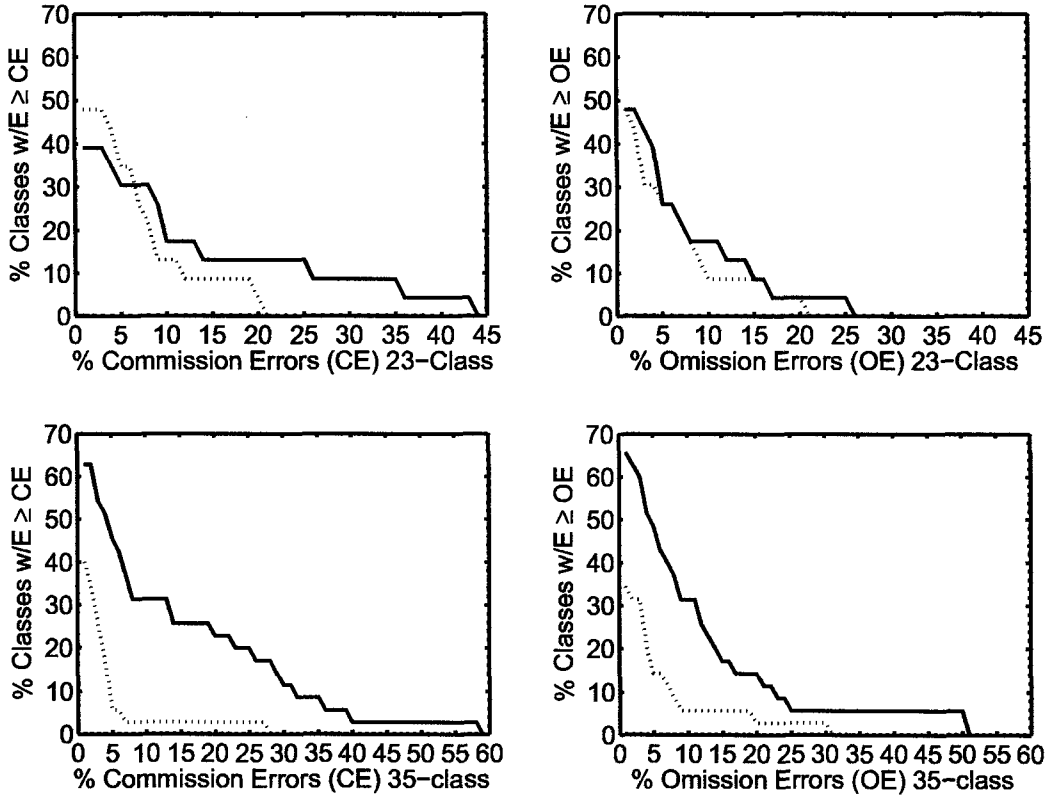


Figure 5.3 : **Left:** The percentage of classes with a commission error greater than a fixed threshold error (E along the x axis) for GRLVQI (dashed) and GRLVQ (solid) for the 23-class problem (top) and 35-class problem (bottom). **Right:** The percentage of classes with an omission error greater than a fixed threshold (E along the x axis) for GRLVQI (dashed) and GRLVQ (solid) for the 23-class problem (top) and 35-class problem (bottom).

Second, in many cases where GRLVQ has significant errors, GRLVQI drastically reduces those errors. There are a few instances when the opposite is true. The extent of the differences is portrayed in Fig. 5.4. Here we graphically show the difference in error, ΔE, between GRLVQ and GRLVQI. $\Delta E = [E(GRLVQ) - E(GRLVQI)]$, where E is either the commission error CE or the omission error OE is plotted on the y axis and the class label placed on the x axis. Positive values indicate GRLVQ has a larger error than GRLVQI and negative values indicate the opposite. For the 23-class problem (Fig. 5.4 top), the difference in the commission and omission errors indicate more error for GRLVQ than for GRLVQI. For the 35-class problem (Fig. 5.4 top), GRLVQI shows even more significantly reduced commission errors (left) and omission errors (right) over GRLVQ.

Although the different error plots presented in Fig. 5.3 and Fig. 5.4 provide some indication as to the significance of the commission error and omission errors, it is difficult to make stronger statements regarding classifier performance based on these plots alone. Table 5.3 presents four numerical values we use to summarize the difference in commission and omission errors between GRLVQ and GRLVQI. We summarize the four methods below.

- **Mean Error:** The average error across all classes. We use this value to compare overall commission and omission errors against the known labeled samples.
- **Mean ΔE** : Based on the difference in error $\Delta E = [E(GRLVQ) - E(GRLVQI)]$, it is the mean of the positive differences $(W+)$ for GRLVQ and the mean of the negative differences $(W-)$ for GRLVQI. Differences of zero are not counted.
- **Signed Ranks:** The sum of the signed ranks of ΔE. This method is discussed in more detail in Section 5.1.4 for the Wilcoxon Signed Ranks Test (WSRT). The signed ranks value for GRLVQ is the sum of the positive ranks $(W+)$ and for GRLVQI it is the sum of the negative ranks $(W-)$. For multiple occurrences of the same differences, the ranks are averaged and distributed among the positive and negative signed ranks. For differences of zero, the rank of each zero is divided evenly between the positive and negative ranks ignoring one outcome if there is an odd number of zeros. We use

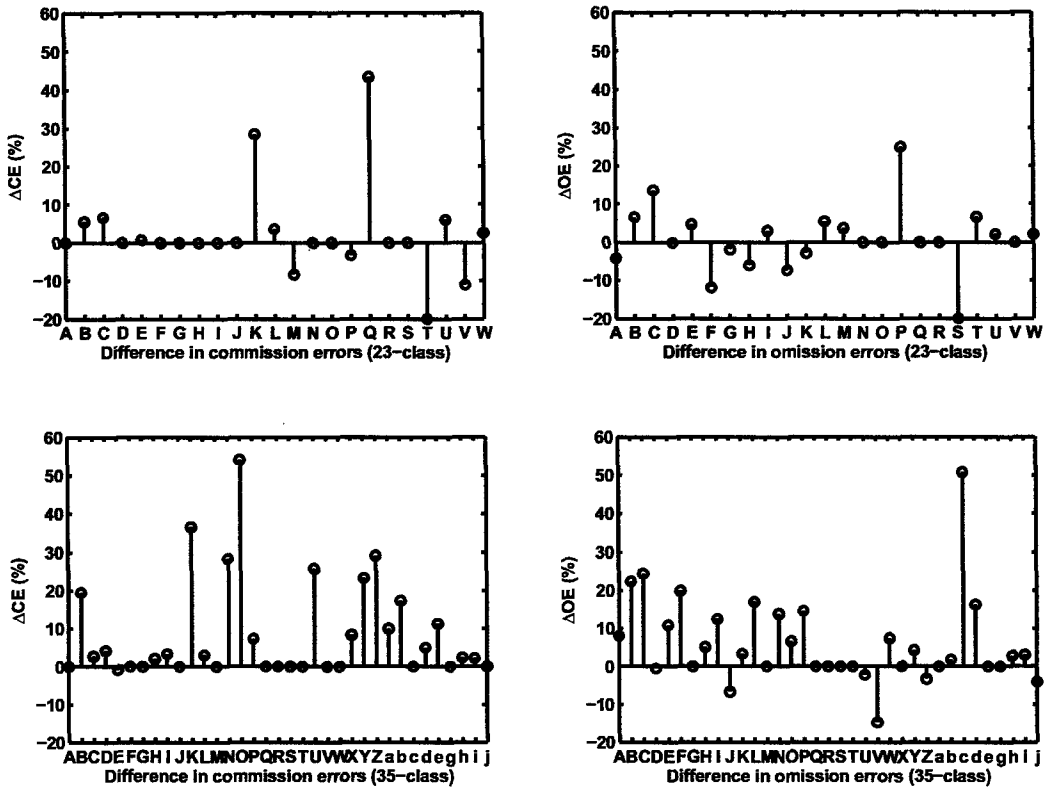


Figure 5.4 : **Left:** The difference in commission errors between GRLVQ and GRLVQI for the 23-class problem (top) and 35-class problem (bottom). **Right:** The difference in omission errors between GRLVQ and GRLVQI for the 23-class problem (top) and 35-class problem (bottom). In all plots along the y axis, the difference in error $\Delta E = [E(GRLVQ) - E(GRLVQI)]$. Positive differences occur when GRLVQ has a larger error than GRLVQI. Negative differences occur when GRLVQI has a larger error than GRLVQ. The x axis in all plots is labeled with the class labels.

the signed ranks to provide a numerical value of the difference in the commission and omission errors. The values will not be used to determine the statistical significance of the results.

- **Sign Count:** The count of positive, negative, and zero ΔE values. With the sign count, zero values are split evenly between GRLVQ and GRLVQI (in the case where there is an odd number of zero values, one is discarded). This method is motivated by the *Sign Test* described in [50]. We use the sign count to provide a numerical value of the difference in the commission and omission errors. The values will not be used to determine the statistical significance of the results.

When comparing the results of the four error measures reported in Table 5.3, one should

			23-Class				35-Class			
			Commission		Omission		Commission		Omission	
			Value	Scaled	Value	Scaled	Value	Scaled	Value	Scaled
Mean		A	6.7%	0.61	4.4%	0.55	10.6%	0.83	8.9%	0.77
Error		B	4.3%	0.39	3.6%	0.45	2.1%	0.17	2.7%	0.23
Mean	$(W+)$	A	12.1%	0.53	7.3%	0.52	14.8%	0.94	12.9%	0.71
ΔE	$(W-)$	B	10.7%	0.47	6.8%	0.48	0.9%	0.06	5.3%	0.29
Signed	$(W+)$	A	242.5	0.88	268	0.97	577.5	0.92	564.5	0.90
Ranks	$(W-)$	B	32.5	0.12	7	0.03	52.5	0.08	65.5	0.10
Sign	$(W+)$	A	13	0.59	12	0.55	27	0.77	24	0.69
Count	$(W-)$	B	9	0.41	10	0.45	8	0.23	11	0.31

Table 5.3 : Results of four summary methods for commission and omission errors. The value column is the actual value of the error method. The scaled column is each reported error scaled by [Value(GRLVQ) + Value(GRLVQI)]. Rows marked with **A** are GRLVQ results and those marked with **B** are GRLVQI results.

read larger values to mean larger error. Generally one would use measures such as the signed ranks and sign count to indicate the significance of classification accuracy where larger values are interpreted as being better. Because we are dealing with errors, larger values are interpreted as being worse. Raw numbers are tabulated in the *Value* column and the scaled form (scaled by [Value(GRLVQ)+Value(GRLVQI)]) in the *Scaled* column. The scaling of the results, although not a common practice in the literature, aides in comparing the results of the four summary measurements based on the commission and omission errors.

For the 23-class problem, the *Mean* ΔE results do not provide sufficient evidence to support the claim of significant improvement of GRLVQI over GRLVQ for either commission or omission errors in the case of the 23-class problem. The remaining three measures (Mean Error, Signed Ranks, and Sign Count), however, are in close agreement that GRLVQI has fewer omission and commission errors for the 23-class problem. The indication with the signed ranks is particularly strong. The results of the four measurements further provide strong evidence that GRLVQ has significantly greater commission and omission errors for the 35-class problem than does GRLVQI. Therefore, based on the analysis of commission and omission errors, we can say that GRLVQI performs better than GRLVQ for the classification

of hyperspectral data.

5.2.3 κ statistic

We use the κ statistic to compare the relative performance between two classifiers measured against pure chance. That is, we interpret the observation of GRLVQI having a larger κ than GRLVQ, as having more confidence in GRLVQIs results over the results obtained by GRLVQ. We feel this is a reasonable approach since κ statistic is normalized, removing that part of the classification resulting from pure chance [23, 60–62]. We use the estimate $\hat{\kappa}$ of Eq. 5.5 to estimate the true κ statistic for the 23-class and 35-class problems.

23-Class Problem

	$\hat{\kappa}_{\text{run1}}$	$\hat{\kappa}_{\text{run2}}$	$\hat{\kappa}_{\text{run3}}$	$\hat{\kappa}_{\text{mean}}$
GRLVQ	0.9685	0.9430	0.9438	0.9518
GRLVQI	0.9895	0.9453	0.9798	0.9715

35-Class Problem

	$\hat{\kappa}_{\text{run1}}$	$\hat{\kappa}_{\text{run2}}$	$\hat{\kappa}_{\text{run3}}$	$\hat{\kappa}_{\text{mean}}$
GRLVQ	0.9076	0.8710	0.9119	0.8968
GRLVQI	0.9828	0.9763	0.9763	0.9785

Table 5.4 : κ statistics for the 23-class problem (Top) and 35-class problem (Bottom).

The results in Table 5.4 indicate that the prediction of GRLVQI agrees better with the known test samples than does GRLVQ for both the 23-class and 35-class problems. Another way of stating it is that there is more confidence in the results obtained by GRLVQI than the results obtained by GRLVQ. This difference appears significant for the 23-class problem and very significant for the 35-class problem. The results of the κ statistic therefore also supports our hypothesis.

5.2.4 Wilcoxon Signed Ranks Test

For our evaluation, the number of trials $N_t = 6$, and in our case GRLVQI outperformed GRLVQ in all six trials (Table 4.1). Results of the positive and negative sum or ranks yields

a $W+ = 21$ and $W- = 0$. Using a table of critical values (see Table II in Wilcoxon [65] and detailed treatment of the table of critical values in McCornack [66]), we find that the probability P, of a rank total T less than $W+ = 21$, is $P = 0.02$ (i.e., $P[T \leq 21 | N_t = 6] \leq 0.02$). This says that the classification accuracy improvement of GRLVQI over GRLVQ for the remotely sensed hyperspectral data in this study are significant to 0.02. Hence, the WSRT too, supports our hypothesis.

5.3 Summary and Discussion

In this chapter we hypothesized that GRLVQI's improvement over GRLVQ for classification of remotely sensed hyperspectral data was significant. First, we used recent theory on the generalization bounds for LVQ-type classifiers and redefined the hypothesis margin based on that which is minimized by GRLVQ(I) similar to previous works on the subject. We show that GRLVQI is expected to generalize better than GRLVQ based on this large margin criterion, which is in agreement with our hypothesis.

Second, the analysis of the commission and omission errors gives one a valuable picture of which classes are difficult to classify and which classes are frequently confused as being a different class. Our analysis of the commission and omission errors provided an second indication that our GRLVQI has better performance than GRLVQ, which is in agreement of our hypothesis.

Third, we calculated the estimate $\hat{\kappa}$ of the κ statistic for each of the three independent jack-knife runs for the 23 and 35-class problems. The κ statistic measured the agreement between the actual and predicted (by the classifier) removing an estimate for chance agreement. The results of the κ statistic calculations are in agreement with our hypothesis.

Finally, to determine if the improved classification accuracy achieved by GRLVQI is statistically significant, we used the Wilcoxon Signed Ranks Test (WSRT) for paired tests.

The results of the WSRT indicate our improved results are significant to a level of $\alpha = 0.02$. With the agreement of all four analysis methods, we conclude that our hypothesis is correct; GRLVQI performs better than GRLVQ for remotely-sensed hyperspectral data.

There are many tests one can use in the evaluation of classifier performance. For the case of comparing only two classifiers, the WSRT is a reasonable choice. One reason the binomial test is not used is because it suffers from several weaknesses [63], one weakness being the binomial test does not take into account the agreement between the two tests. In our review of the literature, it is unclear to us whether or not this same weakness is exhibited by the WSRT since it was never explicitly stated. The κ statistic does not measure the agreement between classifiers but it does give a normalized measure by alleviating the portion due to chance agreement. The point here is that perhaps one should use something like a κ statistic in place of the classification accuracy (or area under the receiver operating characteristic curve) in algorithm comparison tests such as the binomial test, sign test, or the WSRT. One possible issue with using the κ statistic is that the value of κ can be less than zero. Certainly one can view this as meaningless and only perform such tests using the κ statistic iff $\kappa > 0$.

This chapter demonstrated the performance advantage of GRLVQI over GRLVQ for high-dimensional complex data sets such as hyperspectral data. In the next chapter, we use our improved GRLVQI for the joint classification and feature extraction on the wavelet representation of the hyperspectral data and improve upon our already good feature extraction and outstanding classification performance.

Chapter 6

GRLVQI processing in the wavelet domain

In Chapter 4 we showed the superior classification performance of GRLVQI over GRLVQ. In Chapter 5 we used several analysis techniques from the recent literature to show our classification improvements are significant. Based on the results of Chapter 4 and Chapter 5, we use our GRLVQI for the relevance-wavelet model we introduce in this chapter.

In Section 4.4, we stated that the highly correlated nature of the spectral bands may be a limiting factor for further feature reduction using GRLVQI. By applying an appropriate transform to the data, we can alleviate the correlation issue. The goal is not to extract a set of features from the transformed spectra prior to GRLVQI processing, rather transform the spectra into a different feature space and do GRLVQI processing in that feature space.

Since GRLVQI chooses specific coefficients for classification, a sparse transform, in addition to one with decorrelated coefficients, will likely give us additional gains in feature reduction. Further, an efficient transform is desired so as not to incur unnecessary processing costs. The wavelet transform fits the bill given our requirements. It is a data independent transform that has transform coefficients which are "nearly" decorrelated and are sparse. Further, it is an efficient transform computable in linear time. What is unclear at this time, based on the limited success of earlier works [7, 9, 10], is the appropriateness of the wavelet feature space for classification. After reading this chapter, it will become exceedingly clear that the wavelet feature space is indeed a good environment for GRLVQI processing where we obtain superior classification accuracy with a minimal set of features.

Given that we are interested in a transform which has decorrelated coefficients, one might consider Principal Component Analysis (PCA), a data-dependent transform which

has optimally decorrelated coefficients. However, PCA traditionally does not maintain the discrimination capability of the original data, which is especially true for hyperspectral image data. In Section 6.6, we perform a PCA on the 23-class hyperspectral dataset and do GRLVQI processing on the principal components. We demonstrate we can do better using the wavelet representation and provide the classification "sanity check" using PCA only as due diligence, at the end of this chapter.

6.1 The Critically Sampled Discrete Wavelet Transform (CSDWT)

In this section, we provide a cursory look at the Critically Sampled Discrete Wavelet Transform (CSDWT). Interested readers are referred to many quality sources on wavelet theory for further details of this powerful analysis tool (see e.g., Daubechies [67], Burrus et al. [68], Resnikoff and Wells [69], and Vetterli and Hurley [70]).

The wavelet transform has many properties which makes it an ideal analysis tool for a wide variety of problems. First, wavelet coefficients are simultaneously localized in time and frequency and tend to be sparse [71]. Time and frequency localization means you have an idea of what occurred and when it occurred, in contrast to Fourier analysis which gives only perfect frequency localization (you know what occurred, but not when). Wavelet basis functions at different scales are integer shifts and dilations of a single mother wavelet. That is, wavelets have the property of multi-resolution [71]. According to [72,73], wavelets posses the clustering property (i.e., wavelet coefficients adjacent to large (small) wavelet coefficient tend to be large (small)) and wavelet coefficients persist across wavelet scales (that is, large or small values propagate across scales).

The Critically Sampled Discrete Wavelet Transform (CSDWT) represents a signal $f(t)$

as a sum of its scaling coefficients $c(n)$ and wavelet coefficients $d_k(n)$:

$$f(t) = \sum_{n=-\infty}^{\infty} c(n)\phi(t-n) + \sum_{k=0}^{\infty}\sum_{n=-\infty}^{\infty} d_k(n)2^{k/2}\psi(2^k t - n) \tag{6.1}$$

where $\phi(t)$ is the scaling function and $\psi(2^k t - n)$ the wavelet function at scale k.

To perform a k-level CSDWT of the function $f(t)$, one simply takes the inner product of $f(t)$ with scaling function $\phi(t)$ and wavelet function $\psi(2^k t - n)$:

$$c(n) = \int_{-\infty}^{\infty} f(t)\phi(t-n)dt, \tag{6.2}$$

$$d_k(n) = \int_{n=-\infty}^{\infty} f(t)2^{k/2}\psi(2^k t - n), \tag{6.3}$$

where $2^{k/2}$ is a normalizing term.

One can efficiently compute the wavelet and scaling functions using a filter bank (see [74]). Here, the discrete input signal $f[n]$ is filtered with low-pass scaling filter $H(z)$ and high-pass wavelet filter $G(z)$, iterating on the low-pass scaling coefficients at each scale. This process is demonstrated in Fig. 6.1 for a 3-level wavelet transform.

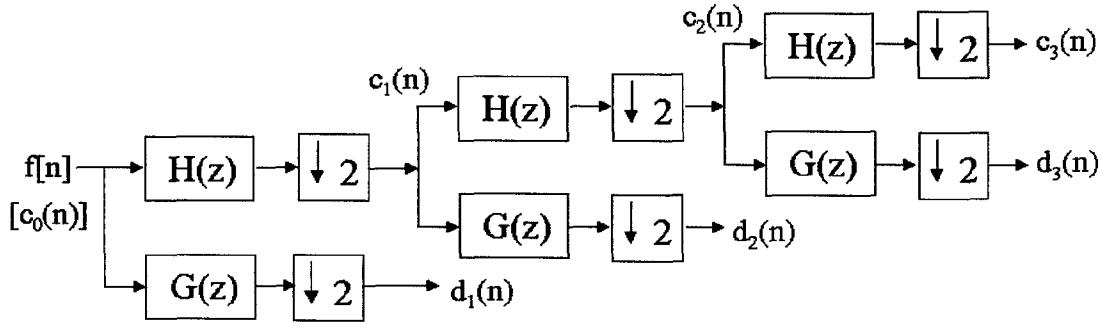


Figure 6.1 : Analysis filter bank for the Critically Sampled Discrete Wavelet Transform. The filters $H(z)$ and $G(z)$ are the z-transforms of the high-pass wavelet filters and low-pass scaling filters respectively. The symbol $\downarrow 2$ denotes a down sampling operation by a factor of two.

When presenting a vector of data for transformation using wavelets, a vector results of possibly different length depending on the implementation of the transform and the original length of the input vector. For a 3-scale CSDWT of our 194-dimensional spectral data using

Daubechies length four orthogonal (Daub4) filters, the resulting transform vector has 200-dimensions. We show the relationship between Fig. 6.1 and the vector of wavelet coefficients at each scale in Fig. 6.2 (this will become important when we analyze our results).

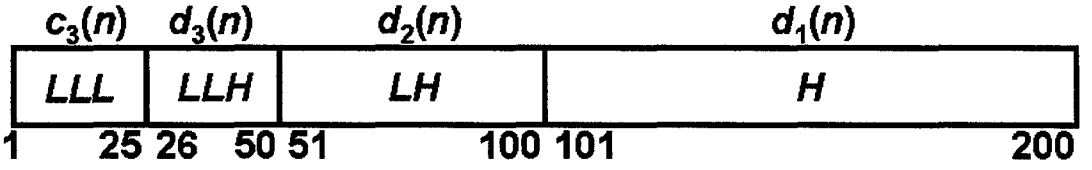


Figure 6.2 : The output from the filter bank of Fig. 6.1 (top) as it relates to the application of the high-pass (H) and low-pass (L) filters (middle) and the range of wavelet coefficients that apply to the specific filter sequence (bottom).

6.2 Wavelet coefficients are "nearly decorrelated"

Before we present results of GRLVQI processing on the CSDWT of the hyperspectral data, we first illustrate what it means for wavelet coefficients to be "nearly decorrelated". The correlation coefficient (ρ) is calculated as:

$$\rho = \frac{\mathrm{COV}(i,j)}{\sqrt{\mathrm{COV}(i,i)\mathrm{COV}(j,j)}}, \tag{6.4}$$

where $0 \leq \rho \leq 1$, COV is the covariance matrix, and the pair (i,j) are the row and column indices. In Fig. 6.3 we plot the magnitude of the matrix of pair-wise correlations (of the original spectral features) for LCVF's 931 labeled samples from the 23-class problem. White pixels correspond to larger ρ where black pixels correspond to smaller ρ. The matrix of correlation coefficients is symmetric with ones along the diagonal (i.e., a coefficient is perfectly correlated with itself).

We see from Fig. 6.4 that the wavelet transform has wavelet coefficients that are nearly decorrelated. By nearly decorrelated we mean most of the pair-wise correlations are small (as indicated by the large number of black pixels in the matrix). There are two major regions that have larger values of ρ. The first region is that part of the signal that is low-pass

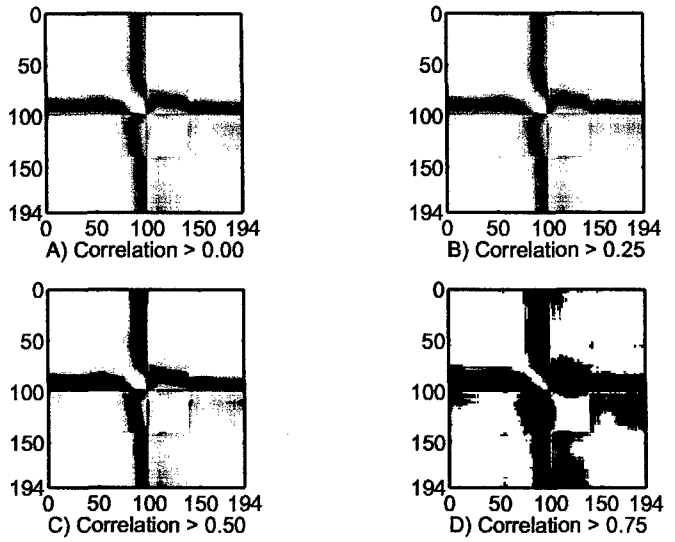


Figure 6.3 : Plot of the correlation coefficient ρ of the covariance matrix for the labeled spectral samples. Lighter pixels correspond to larger correlation coefficients where dark pixels correspond to smaller values. The matrix of correlation coefficients is a symmetric matrix with ones along the diagonal.

filtered only. Recall the low-pass filtered region are the coefficients in Fig. 6.2 marked as LLL and are the scaling coefficients at scale 3 ($c_3(n)$). This satisfies our intuition since the LLL portion of signal is a rough order (or coarse) approximation of the true one-dimensional spectral curve. The second set of regions exhibiting correlation are the boundaries between the wavelet scales (reference Fig. 6.2).

6.3 Experimental setup for GRLVQI processing in the wavelet feature space

In our experiments, we use Daubechies length four orthogonal (Daub4) filters [67]. We iterate the wavelet transform three times generating a 3-level wavelet decomposition of the spectral data. Filter selection and the number of wavelet scales are important considerations which can effect the quality of the features space for classification, the reduction of features, or both. We do not consider filter selection or the number of wavelet scales at this time. Our intent at this point is to investigate the feasibility of the wavelet feature space for GRLVQI

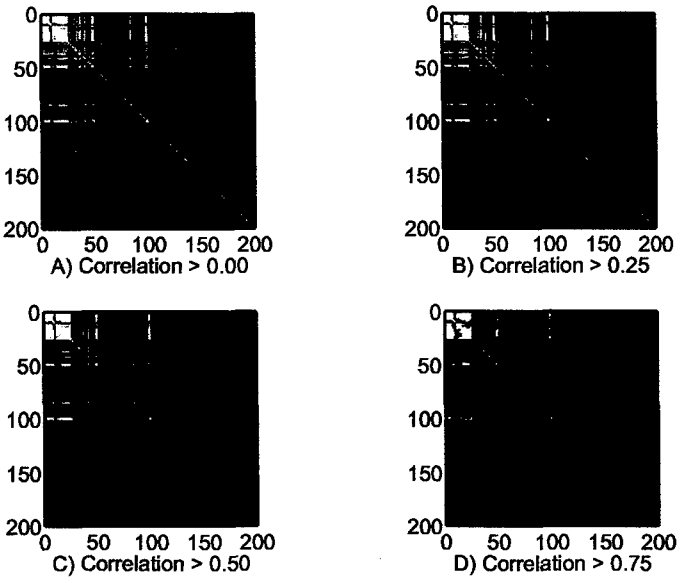


Figure 6.4 : Plot of the correlation coefficients of the covariance matrix for the wavelet transform (using the Daub4 filters) of the labeled spectral samples. Lighter pixels correspond to larger correlation coefficients where dark pixels correspond to smaller values. The matrix of correlation coefficients is a symmetric matrix with ones along the diagonal.

classification and feature extraction.

Our experiments are designed to meet two goals. First, compare the quality of the wavelet feature space to that of the spectral feature space for classification using GRLVQI and feature reduction using thresholding (*Process E* and *Process F* in Table 6.3). Second, evaluate the discrimination and feature reduction capability of the GRLVQI extracted wavelet features to that of the GRLVQI extracted spectral features (*Process D* and *Process B* in Table 6.3) using the Minimum Euclidean Distance (MED) classifier. The MED is of particular importance since it will allow us to impartially compare more typical largest magnitude wavelet coefficient selection as a feature set (*Process C* in Table 6.3) to GRLVQI selected spectral features and GRLVQI selected wavelet coefficients (*Process B* and *Process D*, respectively).

The number of spectral features required to achieve a given classification accuracy for *Process B* is described in Section 4.2. We determine the number of wavelet features required to achieve a given classification for *Process D* in the same manner as described in Section 4.2

Process	Process description
A	Benchmark MED classification with all available spectral features
B	MED classification of the GRLVQI extracted spectral features
C	MED classification of largest magnitude wavelet coefficients
D	MED classification of the GRLVQI extracted wavelet coefficients
E	Classification in the wavelet domain using GRLVQI as the classifier
F	Classification in the spectral domain using GRLVQI as the classifier

Table 6.1 : List of simulations run to compare features discovered by GRLVQI from the spectral data and from the wavelet coefficients.

for spectral features. We add an extra step by taking the inverse wavelet transform on the retained wavelet features prior to MED classification. In principle, the inverse wavelet transform is not required prior to classification as the wavelet transform is linear. However, we perform this extra step for reasons of consistency. As with the MED classification of relevance selected spectral features, relevance is only used in the selection of the wavelet features and *is not* used to scale the data.

As with the relevance selected spectral and wavelet features, the evaluation of the largest magnitude wavelet coefficients (*Process C*) is accomplished similar to that described in Section 4.2, only we select wavelet features in descending order of their magnitudes. That is, we start with a single wavelet coefficient that has the largest magnitude, take the inverse wavelet transform of the test sample set using this single wavelet coefficient, and then classify the reconstructed spectral signature using the MED classifier and tabulate the result. Next we add, to the first wavelet coefficient, the wavelet coefficient with the second largest magnitude. We perform the inverse wavelet transform of the test sample set using these two wavelet coefficients, classify the reconstructed spectral signatures using the MED classifier and tabulate the result. This process continues by adding wavelet coefficients one at a time in descending order of their magnitudes.

6.4 Results of the wavelet feature space for GRLVQI processing

To facilitate the discussion of our results, we will refer to the relevance factors obtained by GRLVQI processing in the spectral domain as *spectral relevance factors.* Similarly, we will refer to the relevance factors obtained by GRLVQI processing in the wavelet domain as *wavelet relevance factors.*

Table 6.2 shows that although GRLVQI in the wavelet feature space (*Process E*) yields only slightly better classification accuracy than GRLVQI in the spectral feature space (*Process F*), executing GRLVQI in the wavelet domain yields significantly fewer features than GRLVQI in the spectral domain. This matches our expectation since wavelet coefficients are (nearly) decorrelated and sparse.

Summary of Classification Results

	A	**B**	**C**	**D**	**E**	**F**
Classification Accuracy	92.8%	96.4%	93.1%	95.7%	97.3%	97.0%
Retained Features	194	52	37	17	17	81

Table 6.2 : Classification accuracy and the corresponding number of significant relevance factors or wavelet coefficients for the six simulations listed in Table 6.3. Processes are grouped according to similarity. *Process A* stands alone, *Process B-D* evaluate the addition of each new feature with the Minimum Euclidean Distance Classifier (MED), and *Process E-F* are GRLVQI results. Tabulated accuracies are computed as the average of the 3-fold cross validation runs. In each run, the overall accuracy is calculated as the average of the individual class accuracies on the test data.

Our feature retention and validation using the MED classifier shows we can achieve 3.6% improvement in classification accuracy using the 52 relevance selected spectral features (*Process B*) discovered with GRLVQI over the benchmark using all (194) spectral features (*Process A*) with a 73% savings in the number of retained features. The largest magnitude wavelet coefficients as features (*Process C*) may be advantageous over relevance selected spectral features (*Process B*) if the given application can tolerate a small degradation in classification accuracy (less than 3%) as it provides an additional 33% savings in features! For relevance selected wavelet features, the MED produces better results than largest mag-

nitude coefficient selection (2.6%) with an addition 54% savings in retained features. The discrimination capability for the MED is slightly better with relevance selected spectral features than with relevance selected wavelet features (0.7%). However, MED classification with relevance selected wavelet features saves an additional 54% in retained features over relevance selected spectral features. In this case, a 0.7% increase in classification accuracy is relatively insignificant compared to the savings in retained features.

Using GRLVQI in the wavelet domain produces the best results (*Process E*) with accuracy slightly better (0.3%) than GRLVQI in the spectral domain (*Process F*) but with nearly 80% savings in the number of retained features. The results in this section clearly demonstrate the potential of our relevance wavelet feature extraction and classification model.

6.4.1 Looking at GRLVQI computed wavelet relevance factors

Relevance learning in the wavelet domain selects the same wavelet coefficients from all wavelet representations of the spectral curves. In Fig. 6.5, we show representative wavelet curves and the computed wavelet relevance factors. We see GRLVQI determines that coarse signal information as most important for classifying our 23-class data set. From Fig. 6.5, we see that coarse signal information is contained in those wavelet coefficients which have been low-pass filtered only (coefficient indices 1-25). This tells us that, for the 23-class and 35-class problems evaluated in this thesis, signal information from the continuum of the spectra provides most of the discriminating information needed by GRLVQI (reference Fig. 3.3). In addition to the continuum, GRLVQI requires only a few details from the *LLH* sub-band (Fig. 6.2, coefficients 26 through 50) to achieve its high-classification accuracy. Although a group of large magnitude wavelet coefficients exist around indices 90-105, GRLVQI does not require them for achieving the desired discrimination of the given classes.

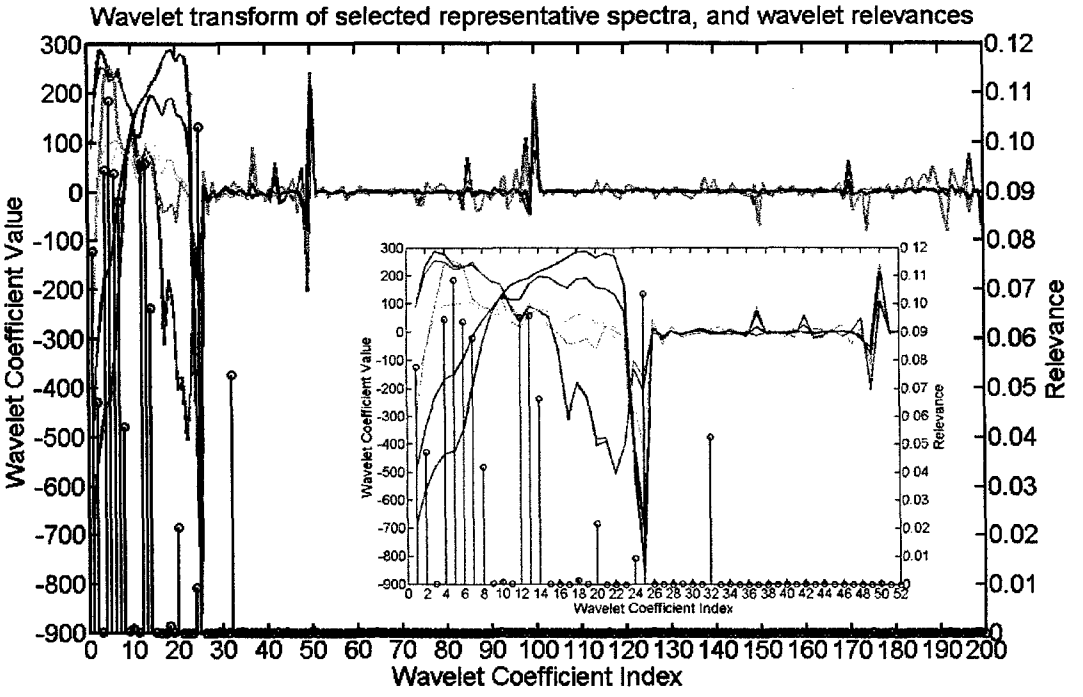


Figure 6.5 : Wavelet transform of class means for classes A (red), G (green), H (orange), L (magenta), O (purple), Q (black), and R (blue) with wavelet relevance factors obtained by GRLVQI (black stem plot) in the wavelet domain. Relevance factors are the averages from three independent runs. Classes were selected to show largest diversity.

6.4.2 Comparing wavelet and spectral relevances

What spectral information is preserved based on the wavelet relevances? Using the inverse wavelet transform on the wavelet relevance factors, we can determine which spectral components are preserved and compare this to the spectral relevance factors from Section 4.3.2.1. For viewing ease, we rescale the inverted wavelet relevance factors using the l_1-norm. Three observations from Fig. 6.6 are worth noting. First, the highly correlated nature of the inverted wavelet relevance factors indicate how well the wavelet transform decorrelates the data for GRLVQI processing. Second, GRLVQI in the wavelet domain places greater emphasis in the 1.4μm to 1.5μm and the 2.2μm to 2.3μm regions than GRLVQI does when processing on the reflectance features. Third, there are some negative values in the inverted wavelet relevance factors. We discuss this third observation in more detail at the end of this

chapter (see Section 6.7).

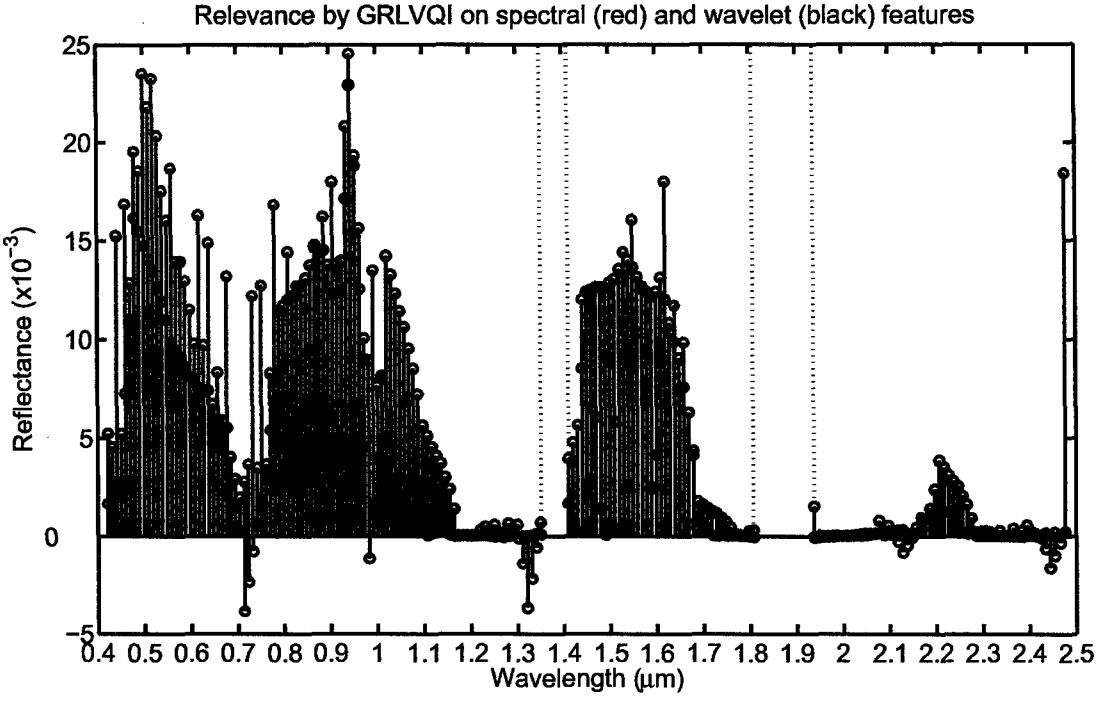


Figure 6.6 : Relevances obtained by GRLVQI in the spectral domain (red) compared to the inverse wavelet transform of the relevances obtained by GRLVQI in the wavelet domain (black). The dotted vertical lines indicate data fallout due to saturation of the water bands.

6.4.3 The energy of retained features

A Parseval's theorem holds for orthonormal wavelets [68], therefore we can compare the energy of the relevant spectral features, relevant wavelet coefficients, and largest magnitude wavelet coefficients (Table 6.2). Total energy is calculated as average pixel energy of the training sample set (to reflect what GRLVQI learns from the data). We define the relevance weighted energy as:

$$E_\lambda = \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{n}\lambda_i\,(x_i^m)^2 \tag{6.5}$$

where i is the dimension index, m the training sample index, M the total number of training samples, and λ_i the relevance factor associated with dimension i. Non relevance-weighted energy calculations use $\lambda_i = 1$ for all i.

We discuss the energy of the retained features to demonstrate there is no consistent relationship between retained energy and achieved classification accuracy. Table 6.3 shows that keeping all available spectral channels has the most energy yet has the weakest MED classification performance (*Process A*). Keeping the wavelet features associated with the 17 largest wavelet relevance factors (*Process D*) has the smallest energy yet is slightly outperformed by keeping the spectral features associated with 52 largest spectral relevance factors (*Process B*) which retains nearly an order of magnitude more energy. Keeping the 37 largest magnitude wavelet coefficients (*Process C*) has the second largest energy yet is outperformed by the 52 SF and 17 WF cases. Interestingly, the 17 retained and scaled wavelet features have ≈3.3 times more energy than the 52 retained and scaled spectral features, yet has slightly worse classification performance.

Average Retained Energy per Pixel for MED Classification

	w/o relevance	w/relevance	Accuracy
52 SF	5.89×10^{6}	1.45×10^{3}	96.4%
17 WF	6.91×10^{5}	4.12×10^{3}	95.7%
37 WC	9.61×10^{5}	NA	93.1%
194 SF	2.49×10^{7}	NA	92.8%

Table 6.3 : The retained energy for the 52 spectral features (52 SF) corresponding to the 52 largest spectral relevance factors, the 17 wavelet features (17 WF) corresponding to the 17 largest wavelet relevance factors, the 37 largest magnitude wavelet coefficients (37 WC) retained using more typical wavelet processing, and for all spectral features (194 SF).

6.4.4 Comparing GRLVQI selected wavelet features to largest average magnitude coefficients and their standard deviations

One might anticipate that GRLVQI would "discover" those wavelet indices corresponding to the largest magnitudes or the largest variances over the entire training set as most important for classification. In this section, we show that GRLVQI learns something different. The mean of all wavelet transformed signatures for the 23-class problem is the black curve in Fig. 6.7 top. The 17 largest magnitude wavelet coefficients are the red stems and the 17 largest wavelet relevance factors are the blue stems. Coefficients corresponding to both

largest magnitude and largest wavelet relevance factors are indicated by black stems. Similarly, the standard deviation of all wavelet signatures is shown as the black curve of Fig. 6.7 (bottom). The 17 largest standard deviations of the wavelet coefficients are indicated by the red stems and wavelet coefficients corresponding to the 17 largest wavelet relevance factors are the blue stems. Wavelet coefficient indices included in both the largest standard deviations and the largest wavelet relevance factors are indicated by the black stems. Clearly, although there is overlap in both plots, the largest relevances assigned by GRLVQI to wavelet coefficients are not the same as the wavelets with largest magnitudes or largest variances.

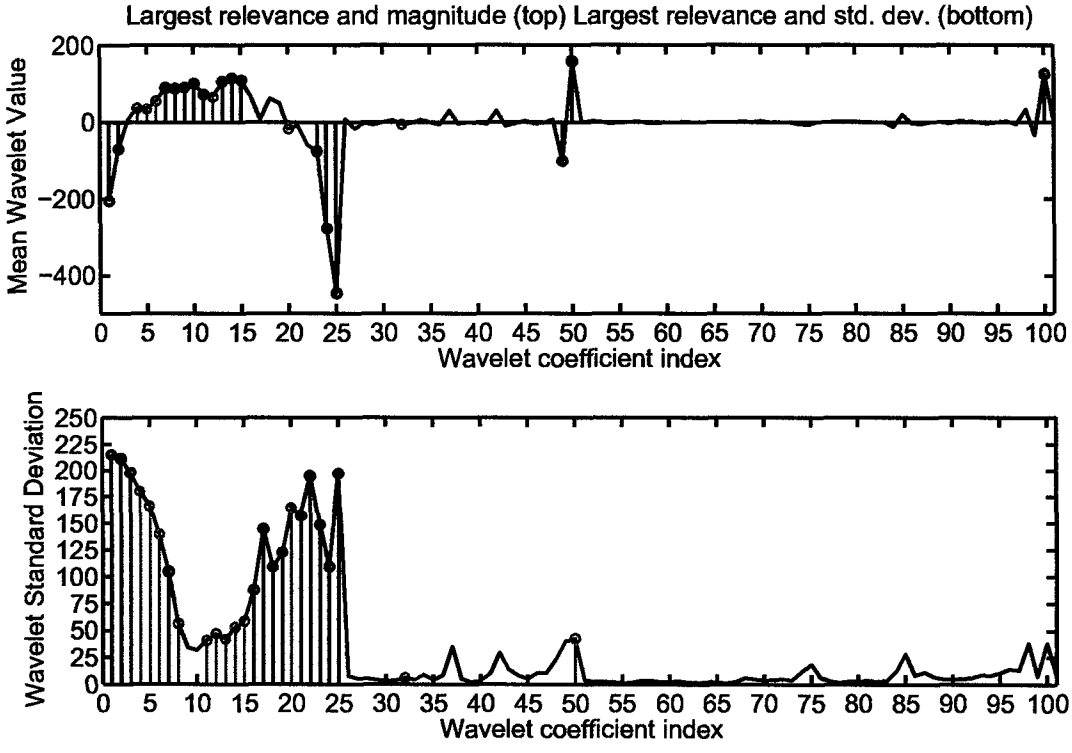


Figure 6.7 : **Top:** Comparison of the 17 largest magnitude wavelet coefficients (red stem plot) with the 17 wavelet coefficients corresponding to the 17 largest wavelet relevance factors (blue stem plot.) The black curve is the mean wavelet representation for all training samples. The black stem plot shows the indices for which the corresponding wavelet belongs to both red and blue sets. **Bottom:** Comparison of the 17 largest standard deviations of the wavelet coefficients (red stem plot) with the 17 wavelet coefficients corresponding to the 17 largest wavelet relevance factors (blue stem plot.) The black curve is the standard deviation of the wavelet representation of all training samples. The black stem plot shows the indices for which the corresponding wavelet belongs to both red and blue sets

6.4.5 Discussion of discontinuities in spectral data

Deleting image bands associated with irrecoverable spectral data causes discontinuities in the spectra. The most common example in remote sensing spectral of bad data are the two wavelength windows (see Fig. 6.8 top) where the atmospheric water vapor saturates the instrument response. A commonly accepted method for dealing with these two regions is to simply delete them. This process results in a piecewise signal with discontinuities at the boundaries (Fig. 6.8 bottom).

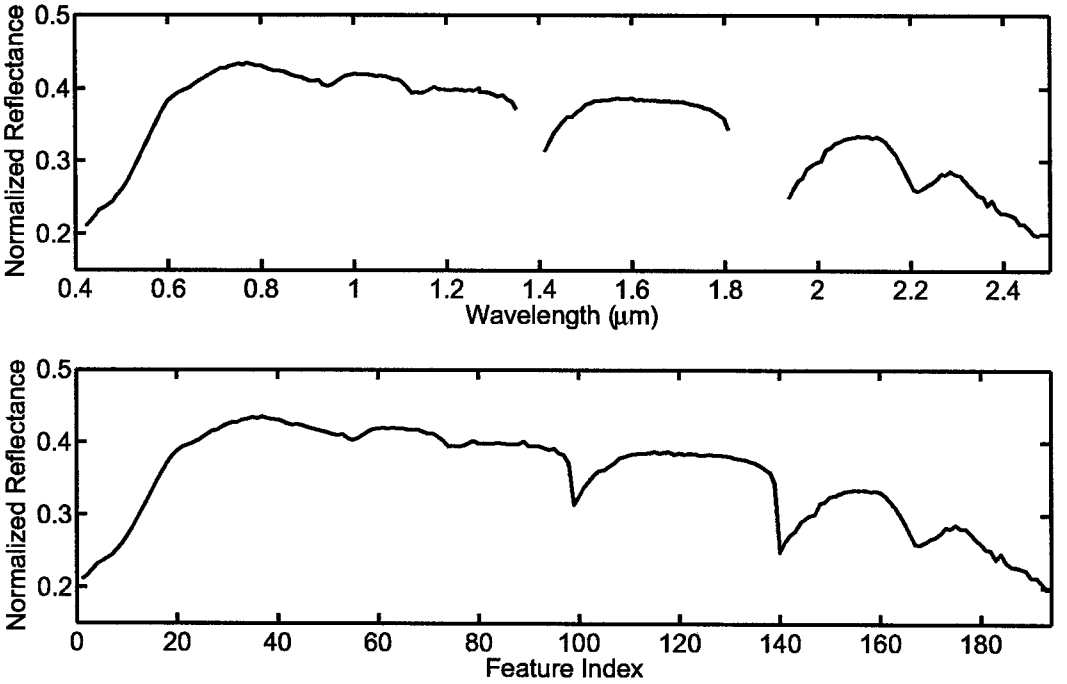


Figure 6.8 : Spectral discontinuities caused by data fallout. **Top:** Normalized reflectance spectrum vs. wavelength with "empty" regions where data are deleted due to saturation of the atmospheric water bands. **Bottom:** Reflectance vs. feature index illustrating the band locations of the discontinuities as they appear to the wavelet transform.

Deleting the spectral bands has no ill effect on our ability to process the data using GRLVQI in the spectral domain. However, these discontinuities can manifest in the wavelet transform coefficients creating a set of false features. These false features are a concern if using GRLVQI because it may waste relevance resources by learning artifacts. Although

we have successfully demonstrated the quality of the wavelet features space for classifying hyperspectral data with the discontinuities present, we feel we can do better using a different wavelet transform with special properties which may help mitigate the effects of the discontinuities. In Section 6.5 we consider the Dual-Tree Complex Wavelet Transform (DTℂWT).

6.5 The Dual-Tree ℂomplex Wavelet Transform (DTℂWT) to remedy data discontinuities

We investigate the application of the Dual-Tree ℂomplex Wavelet Transform (DTℂWT) to decompose the spectral data in an effort to mitigate the effects of the discontinuities resulting from deleted image bands. The magnitude of the DTℂWT has several desirable properties which we may be able to leverage. First, it has reduced oscillatory effects when a discontinuity is encountered [75]. Other potentially useful properties include near shift invariance and reduced aliasing effects during reconstruction if wavelet coefficients are modified during processing [75] (e.g., filtering, quantization, etc).

Section 6.5.1 provides a cursory discussion of the DTℂWT. In Section 6.5.2 we demonstrate the reduced oscillatory effects of the magnitude of the DTℂWT compared to the CSDWT with the Daub4 filters. We do GRLVQI processing on the four components of the DTℂWT (real, imaginary, magnitude, and phase) at two different wavelet scales for the 23-class problem in Section 6.5.3 and conclude that the imaginary part is most useful for our application due to odd-symmetry of its basis functions, as discussed in Section 6.5.4. We then use the Odd-Symmetric Discrete Wavelet Transform (OSDWT) (i.e., the imaginary component of the DTℂWT) for the remainder of the evaluation. Based on the increase in accuracy and feature reduction from the 3-scale to 4-scale OSDWT, we extend the 23-class problem results to a 3-scale, 4-scale, and 5-scale CSDWT and OSDWT.

Section 6.5.5 evaluates GRLVQI processing on the a 3-scale, 4-scale, and 5-scale CSDWT and OSDWT for the 35-class problem. Finally, in Section 6.5.6, we use the MED to assess

the GRLVQI features for the 23-class problem. We do not extend the MED assessment to the 35-class problem because the results are much the same and do not offer any further insights.

6.5.1 Background on the DTℂWT

The DTℂWT (see Selesnick et al. [75]) defines two trees (or filter banks, see Fig. 6.1): one filter bank computes the real part while a second filter bank computes the imaginary part. Each low-pass scaling coefficient and high-pass wavelet coefficient is the sum of its real and imaginary pieces: $c^c(n) = c^r(n) + jc^i(n)$, $d_k^c(n) = d_k^r(n) + jd_k^i(n))$, where $j = \sqrt{-1}$.

In the case of the real component, the scaling function $\phi(t)$ and wavelet function $\psi(t)$ are both real and even (symmetric) [75]. Conversely, for the imaginary component, the scaling and wavelet functions are both imaginary and odd (anti-symmetric) [75]. We can similarly write the scaling and wavelet functions as the sum of their real and imaginary parts: $\psi^c(t) = \psi^r(t) + j\psi^i(t)$ and $\phi_k^c(t) = \phi_k^r(t) + j\phi_k^i(t)$.

6.5.2 The effects of the CSDWT and DTℂWT on discontinuities

To demonstrate the reduced oscillatory effects of the DTℂWT over the CSDWT, we define a function to place discontinuities in the vicinity of the locations where spectral bands are deleted. For the LCVF data set, the missing data lies "between" the band pairs (98,99) and (139,140) (Fig. 6.8 bottom). We define our function as

$$f[n] \;=\; u[n-97] - u[n-101] + u[n-138] - u[n-142]. \tag{6.6}$$

where n indicates band index and $u[\cdot]$ is the unit step function. We then take the wavelet transform of $f[n]$ using the CSDWT with the Daub4 filters (shown in Fig. 6.9 top) and the DTℂWT (shown in Fig. 6.9 bottom). Fig. 6.9 shows that the DTℂWT results in less oscillation of the wavelet coefficients.

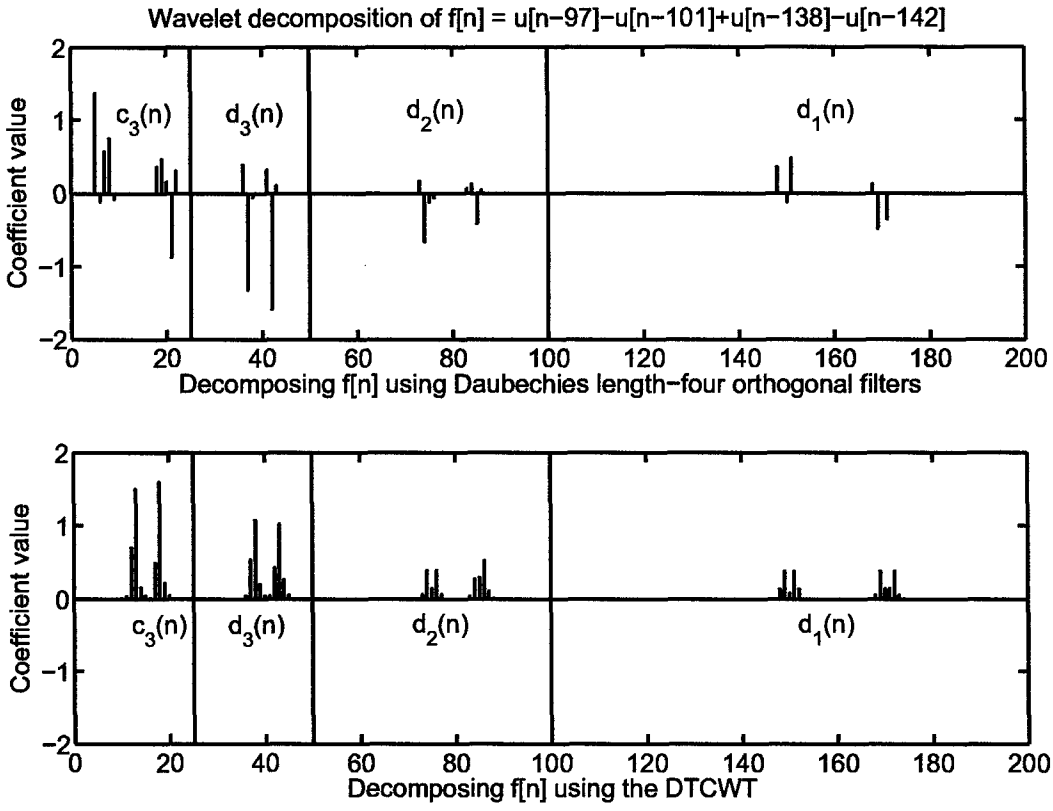


Figure 6.9 : Wavelet transform of $f[n]$ in Eq. 6.6 in the vicinity of the spectral discontinuities. **Top:** Three scales of the CSDWT of $f[n]$ using the Daub4 filters. **Bottom:** The magnitude of 3-scales of the DTℂWT of $f[n]$.

6.5.3 Evaluating the DTℂWT on the 23-class LCVF problem

Our initial hypothesis was that GRLVQI processing on the magnitude of the DTℂWT would yield best results in the face of discontinuities in the data (and consequently, artifacts in the wavelet coefficients). However, based on the results presented in Table 6.4, the imaginary component of the DTℂWT (especially for the 4-scale transform) shows superior results. The imaginary component of the DTℂWT has odd basis functions and will be referred to as an Odd-Symmetric Discrete Wavelet Transform (OSDWT). We provide some discussion on why the OSDWT yields superior results in the next section (Section 6.5.4).

Table 6.4 shows that the quality of the wavelet feature space for both classification and feature extraction is dependant on the number of scales used to decompose the signal.

		Real	Imaginary	Magnitude	Phase
3-level	Acc (%)	97.35%	97.44%	95.99%	74.56%
DTℂWT	# Features	55	16	51	70
4-level	Acc (%)	95.13%	98.28%	83.50%	69.97%
DTℂWT	# Features	15	11	95	75

Table 6.4 : Accuracy and number of features from GRLVQI in the DTℂWT domain. Features with relevances ≥ 0.001 were counted.

To gain further insights, we consider GRLVQI processing on a 3-level, 4-level, and 5-level decomposition of the spectral data using both the CSDWT and the OSDWT (Table 6.5).

Although the best accuracy is achieved with the 4-level CSDWT, the best feature reduction is achieved with the 4-level OSDWT. Given that a 0.2% difference in classification accuracy for a problem with 310 samples is insignificant, the best tradeoff between retained features and classification accuracy is clearly with the 4-level OSDWT. Here we achieve similar classification performance with 15 features (or 7.2%) from the OSDWT while the CSDWT requires 24 features (or 11.5%). One should observe from Table 6.5 that there is no consistent relationship between classification accuracy and feature extraction performance versus the number of scales of the wavelet transform. If any, the 4-level decomposition for both the CSDWT the OSDWT is best. Although we may hypothesize that a 4-level wavelet decomposition produces best results, an in-depth study is required before we can show this hypothesis correct (a topic for continued research).

	CSDWT			OSDWT		
	3-level	4-level	5-level	3-level	4-level	5-level
Accuracy	97.3%	98.2%	97.0%	97.9%	98.0%	96.9%
Features	17	24	27	18	15	30

Table 6.5 : **23-Class Problem:** Accuracy and number of features for GRLVQI processing in the wavelet domain. Features with relevances ≥ 0.001 were counted.

What signal information is retained by the OSDWT? Looking at the wavelet relevance factors in Fig. 6.10, we see that GRLVQI focuses on low-frequency signal information (i.e., the continuum of the spectra). Based on the results of the CSDWT, this comes at no surprise.

The main difference between the GRLVQI computed relevances for the CSDWT (Fig. 6.5) and the those computed for the OSDWT is an additional *very* significant relevance factor outside of the low-pass region. This is important since it indicates that GRLVQI is able to reduce the amount of low-frequency signal information (i.e., less emphasis on the continuum) for classification while placing emphasis on more high-frequency signal information. The end result is fewer wavelet coefficients for the same classification accuracy.

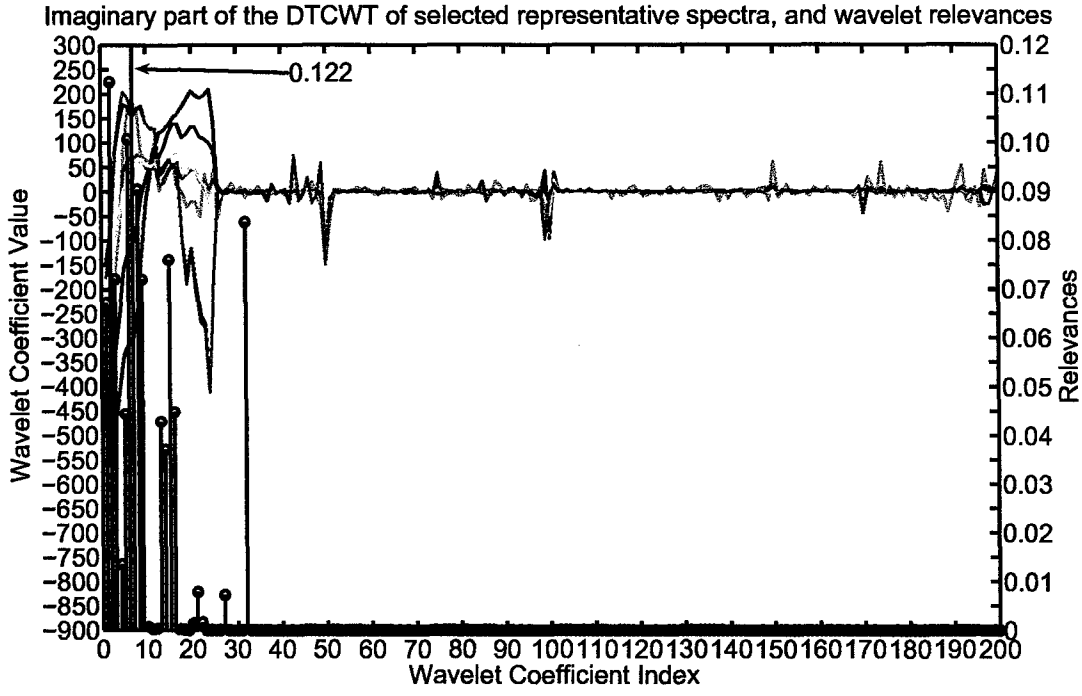


Figure 6.10 : The Odd-Symmetric Discrete Wavelet Transformation of class means for classes A (red), G (green), H (orange), L (magenta), O (purple), Q (black), and R (blue) with wavelet relevance factors obtained by GRLVQI (black stem plot) in the wavelet domain. Relevance factors are the averages from three independent runs from the 23-class problem. Classes were selected to show largest diversity.

6.5.4 Odd-Symmetric Discrete Wavelet Transform (OSDWT) and its importance on spectral feature identification

As discussed in Section 6.5.1, the real component of the DTCWT is even and real while the imaginary component is odd and imaginary. When projecting a function on even wavelet bases, edge responses (i.e., high-frequency information) result in zero-crossings at edge loca-

tions making it more difficult to pinpoint the exact location of the edges. When projecting a function onto odd wavelet bases, edge responses result in a peak in the wavelet domain. This accentuates the differences at the discontinuities and makes singularity identification much easier [B. Johnson, private communication].

Discriminating absorption bands in hyperspectral data are often narrow shapes with sharp boundaries that must be preserved. For even wavelet basis, the important features result in zero-crossings at their locations making it more difficult for GRLVQI to use these features for class discrimination. In contrast, odd wavelet basis functions identify the sharp boundaries with a peak in the wavelet domain. The consequence is better classification accuracy with the imaginary component than with the real component or with the magnitude of the complex wavelet and scaling coefficients. We hypothesize that a wavelet system based on the CSDWT, using odd-symmetric wavelet basis functions, would similarly allow GRLVQI to achieve improved classification accuracy and feature reduction performance.

6.5.5 GRLVQI processing on the 35-class problem decomposed using the OSDWT

To further evaluate and compare the OSDWT to the CSDWT with the Daub4 filters, we extend our study to the more difficult 35-class problem discussed in Section 3.2. The 35-class problem adds additional sub-class structure to the 23-class problem in the form of high-frequency dips and bumps in the spectra. If the high-frequency information in the spectral curve is truly the important distinguishing spectral features, then we should expect that GRLVQI processing with the OSDWT to have better classification and/or features extraction capability for the 35-class problem as compared to the CSDWT.

Classification accuracies and the number of retained features are tabulated in Table 6.6 for the CSDWT and the OSDWT. We perform a 3-level, 4-level, and 5-level wavelet decomposition using the wavelet transform and process the wavelet representation of the 35-class

data with GRLVQI. Here we see a much clearer advantage of the OSDWT over the CSDWT for each wavelet representation. Best classification accuracy results are with a 3-level decomposition using the OSDWT. Once again, however, the feature reduction using the OSDWT at 4-scales is best. The 0.5% accuracy loss between 3-scales and 4-scales is still relatively insignificant for a test sample pool of 488 samples and 35-classes considering the gain in feature reduction is much more significant. With the 4-level OSDWT, only 16 wavelet coefficients are required for classification, 24% fewer than for the 3-level transform using the OSDWT.

	CSDWT			OSDWT		
	3-level	4-level	5-level	3-level	4-level	5-level
Accuracy	95.7%	95.6%	95.4%	97.4%	96.9%	95.9%
Features	26	26	33	21	16	22

Table 6.6 : **35-Class Problem:** accuracy and number of features for GRLVQI processing in the wavelet domain. Features with relevances ≥ 0.001 were counted.

6.5.6 MED assessment of the discrimination capability of the GRLVQI selected features from the OSDWT

We once again visit the Minimum Euclidean Distance (MED) classifier for an independent assessment of the discrimination capability of the wavelet features GRLVQI deems relevant. The process for selecting wavelet coefficients is precisely the same as discussed previously in Section 6.3. The best classification and the number of wavelet coefficients required to obtain that classification is tabulated in Table 6.7. As a baseline, recall the MED classification with all available features is 92.8% (see Table 6.2) for the 23-class problem (results are not duplicated in Table 6.7).

Three general observations can be made from Table 6.7. First, the GRLVQI-selected features (wavelet coefficients) lead to a better classification than using all available features. Second, the wavelet representation by the OSDWT yields consistently similar classification results to the CSDWT with the Daub4 filters. Third, fewer features are required for best

	CSDWT			OSDWT		
	3-level	4-level	5-level	3-level	4-level	5-level
Accuracy	95.7%	96.2%	95.8%	96.2%	95.7%	96.2%
Features	17	14	14	12	10	14

Table 6.7 : Classification accuracy and corresponding number of features for the Minimum Euclidean Distance Classifier for the 23-class problem.

MED performance with the OSDWT, than with the CSDWT using the Daub4 filters.

6.5.6.1 Discussion on the effects of discontinuities in the wavelet domain

Both the 23-class and 35-class problems benefited from the OSDWT. What is the effect of the jump discontinuities in the wavelet domain? One interpretation is that the samples in each class have similar jump discontinuities, which are not exhibited by other classes. Since odd wavelet basis have heightened edge responses making singularity identification easier, these discontinuities may in fact be good distinguishing features for classification.

A second interpretation of the effects of the discontinuities is that they contaminate the same coefficients to some degree. Because the false features are consistent, GRLVQI does not learn them as differences and hence they are ignored. A third interpretation is that the contamination is not significant and appears as noise to GRLVQI. Since LVQ-type classifiers are impervious to noise, the noisy coefficients do not have a negative affect on classification or feature reduction performance.

6.6 Principal Component Analysis

Principal Component Analysis (PCA) is a popular technique for feature reduction. It has the important property that principal components are maximally decorrelated. This fits well with our need for a transformation of the data. One problem with PCA is its dependence on the training data requiring a minimum of $n+1$ samples (n being the number of dimensions of

the data) to (eventually) create the transformation matrix. For hyperspectral data where the number of bands may be several hundred, this requirement can be prohibitive, especially in the case were one is interested in classifying rare materials which have few training samples. Even in the case where one has an adequate number of training samples, the PCA is not likely a good option for hyperspectral data.

6.6.1 Principal Component Analysis – A brief explanation

When considering PCA, it would be ideal to have the same number of samples from every class so that each class may have an equal representation when calculating the principal components. However, this is often not the case, as with our problem sets. To compute the Principal Components, one can follow series of simple steps. First, subtract the mean of each input dimension. Second, compute the covariance matrix of the zero-meaned training data

$$\text{COV} = E\left[(X - E[X])(X - E[X])^{\mathrm{T}}\right] \tag{6.7}$$

where X is a matrix with each observed spectral signature is a row entry. Since we zero mean the data, $E[X] = 0$. Third, compute the eigenvalues and eigenvectors of the covariance matrix and order eigenvectors in descending order of their eigenvalues. Fourth, compute the desired number of principal components by projecting the data onto the corresponding number of eigenvectors. The projection of the data onto the eigenvector corresponding to the largest eigenvalue is the first principal component. The second principal component is the projection of the data onto the eigenvector corresponding to the second largest eigenvalue. This process of projecting data onto the eigenvectors continues until one has the desired number of ordered features.

6.6.2 GRLVQI on the principal components on the LCVF data

Recall we use a 3-fold cross validation to evaluate classifier performance. In PCA, we use the training data from each fold to compute the matrix of eigenvectors used to compute the principal components of the entire data set. This gives us three different principal component representations of the data for which we do GRLVQI processing. Consistent with results presented thus far, the accuracy for each fold is computed as the equal-weighted class accuracy (Section 4.1) and we average the results across all three folds.

Although GRLVQI does not deem the first N principal components as most important for classification (see Fig. 6.11), it does place a significant amount of the relevant resources in this range. For example, the first 6 principal components have a relevance sum of 0.63 and a relevance sum of 0.84 for the first 20 principal components. We see many spurious (yet significant above a threshold of 0.001) beyond the 150^{th} principal component contributing 0.12 of the total relevance. Using a threshold of 0.001, we find 60 significant principal components. As we expect, using the principal components as our feature space does not provide the necessary class discrimination capability indicated by the 71.3% classification accuracy achieved by GRLVQI for the 23-class problem.

6.7 Summary

This chapter introduced a new wavelet coefficient selection paradigm to select those wavelet coefficients *important* for classification. It is unique in that the relevance-wavelet model linearly selects wavelet coefficients for classification. This phase of the research began using the Critically Sampled Discrete Wavelet Transform (CSDWT) using Daubechies length four orthogonal (Daub4) filters to provide a sparse representation of the data with coefficients which are nearly decorrelated. However, due to jump discontinuities introduced into

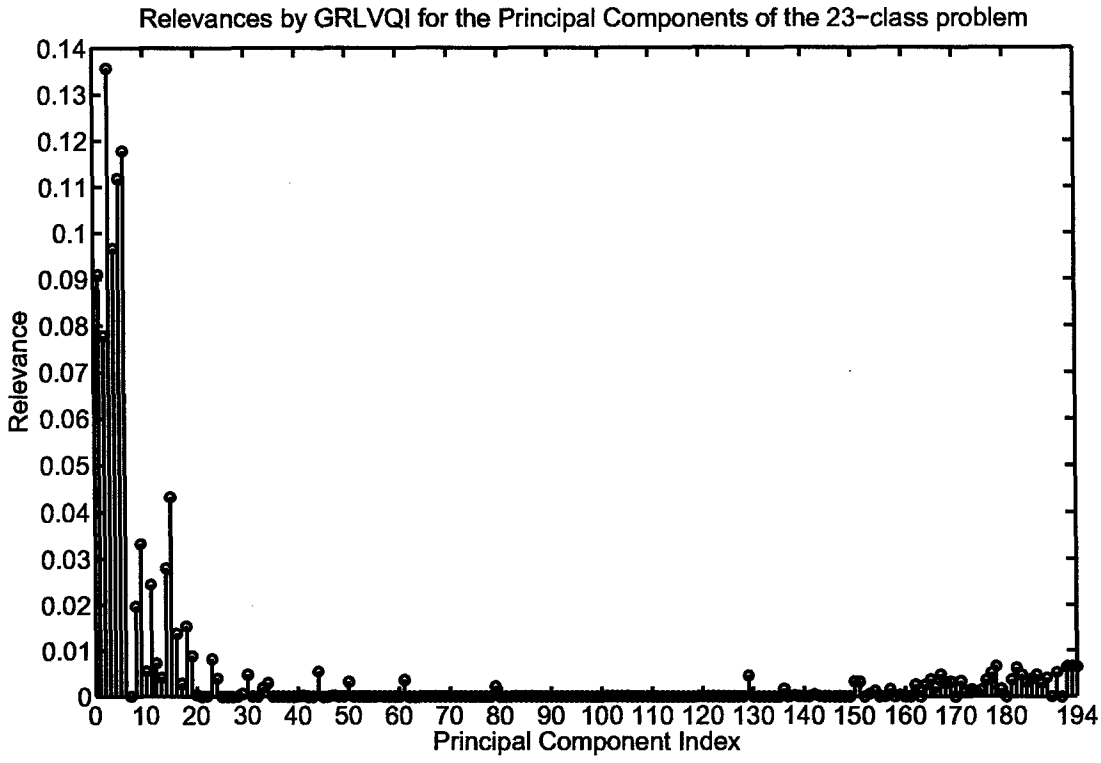


Figure 6.11 : Relevances computed by GRLVQI for the principal components of the LCVF data.

the spectral signatures from deleting bands with bad data, we believed the CSDWT to be sub-optimal because the high-frequency false features would contaminate the wavelet coefficients having a rippling effect with additional scales of the transform. We surmised a better wavelet representation would bring about either a reduction in the required number of wavelet coefficients or an increase in classification accuracy, or both. This lead to the investigation of the Dual-Tree Complex Wavelet Transform, where we discovered that the odd wavelet basis functions yield better results than the even basis functions of the real part. Although we hypothesized that an Odd-Symmetric Discrete Wavelet Transform (OSDWT), in general, would be better suited for subsequent GRLVQI processing for feature extraction and classification, the verification of this hypothesis is a topic for future research.

We have successfully leveraged the sparseness and decorrelation properties of the wavelet transform for classification and feature extraction with resounding success where others have

fallen short. Furthermore, we evaluated our relevance-wavelet model, not on simple pedagogical classification problems, rather on very complex high-dimensional data and produced exceptional results. We were able to improve our already outstanding classification accuracies compared to what GRLVQI can achieve in the spectral domain by selecting a form of the wavelet transform that has additional desirable properties that better fit anomalies in our data. While the feature reduction in the spectral domain was quite good, our relevance-wavelet model dramatically reduces the number of retained features to just a small fraction of the overall number of available features.

Chapter 7
Summary and Discussion

7.1 Summary

In this thesis, we investigated the doubly adaptive neural learning paradigm of Generalized Relevance Learning Vector Quantization (GRLVQ) for joint classification and feature extraction for remotely sensed hyperspectral image data. We presented an original analysis of the LVQ2.1 and G(R)LVQ windows and came to the important conclusion that restricting the influence of the decision boundary to samples that lie in the mid-region between the in-class and out-of-class winning prototype vectors is not necessarily the right choice for class boundary definition. During our analysis, we found that GRLVQ has the potential for diverging prototype vectors and suffered from poor prototype utilization. Our contributions to relevance learning, captured in our GRLVQ-Improved (GRLVQI), solves the divergence problem, ensures good prototype utilization, and increases classification accuracy.

In the careful design of our evaluation process and the design of the GRLVQ and GRLVQI classifiers, we consulted the literature to ensure we adhered to best practices. We used a 3-fold cross validation to obtain independent classification results and used the equal-weighted classification accuracy to measure classification success. We further summarized the literature relating to the design and initialization of LVQ-type classifiers where we looked into the theory on generalization bounds to guide the number of prototype vectors assigned to each class. This investigation revealed that the current theory (only valid for binary classification problems) only gives guidance on the total number of prototypes for the classifier and, as a result, was of little use for this design consideration. For prototype initialization, we found that, due to conscience learning, a uniform random initialization worked well for

our problem. Finally, during the trial-and-error process for setting classifier learning parameters, we confirmed the general consensus that the classification results were not overly sensitive to parameter values and that it was imperative one decayed learn parameters with increased training time.

The additional power of GRLVQI, relative to GRLVQ, is not always warranted. For easier classification problems, we believe that GRLVQI is not necessarily likely to produce better classification or feature extraction results than GRLVQ. However, based on GRLVQI's increased classification results for the Lunar Crater Volcanic Field (LCVF) hyperspectral data set, we hypothesized that GRLVQI performs better than GRLVQ for sufficiently complex high-dimensional data. This claim was validated using analysis techniques commonly used in the remote sensing community (commission/omission errors and the κ statistic) as well as other theoretical and statistical methods presented in the literature (theory on generalization bounds and the Wilcoxon Signed-Ranks Test). In revisiting generalization bounds, we found that recent works by Hammer et al. [47] define the empirical margin in a manner which is inconsistent with earlier works by Crammer et al. [14]. That is, the margin definition in [47] is not what is maximized by GRLVQ in [1] or our GRLVQI. By redefining the margin to that which is maximized by GRLVQ(I), we find that the definition of the empirical loss based on the correct definition of the margin shows that GRLVQI is expected to generalize better than GRLVQ. These results agree with the remaining three methods of performance evaluation we considered in Chapter 5. From this point onward in our research, we used our improved GRLVQI.

From the results of GRLVQI processing in the spectral domain, we found it appeared to be limited in the amount of feature reduction that could be obtained due to the correlated nature of the hyperspectral data. We proposed a new model for classification-driven feature extraction that combined the power of our contributions to neural-relevance learning with

the power of the wavelet transform. Due to wavelet transform properties, specifically that the wavelet coefficients are sparse and nearly decorrelated, our model results in exceptional classification performance with a significantly reduced feature set. By doing GRLVQI processing on the principal components of the spectral data, we demonstrated that we need something more in a transform than decorrelated coefficients to obtain best classification and feature extraction performance.

Although transforming the data to the wavelet domain and doing GRLVQI processing in that domain showed excellent results, we re-evaluated the data to see if it exhibited properties that might further influence the chosen transform. One seemingly obvious issue was that of discontinuities caused by deleting spectral bands with irrecoverable data. Our belief was that the unnaturally occurring discontinuities would affect the ability of GRLVQI to gain the best possible classification while extracting a minimal set of features. This lead to an investigation of the Dual-Tree Complex Wavelet Transform (DTCWT) because it has the additional desirable property of reduced oscillation (or ringing) of the wavelet coefficients in the face of data discontinuities. Of the four pieces of information from a complex signal (real and imaginary parts and the magnitude and phase) we found that processing on the imaginary component of DTCWT produced best results. We surmised the imaginary component of the DTCWT (or more generally, an Odd-Symmetric Discrete Wavelet Transform (OSDWT)) produces best results because its basis functions better identify the distinguishing spectral characteristics than the even symmetry of the real-component, or the Critically Sampled Discrete Wavelet Transform (CSDWT) using the neither even nor odd Daubechies length-four orthogonal (Daub4) filters.

In our investigation of wavelets, we find that low-frequency signal information provides a majority of the important features GRLVQI requires for generating a given classification. In contrast, little high-frequency information is used. We believe that these high-frequency

coefficients are the minor "details" GRLVQI requires to achieve the superior classification and feature reduction performance demonstrated in this thesis.

7.2 Novelty of the relevance-wavelet model for classification

The novelty of our relevance-wavelet model for feature extraction and classification is not to be understated. Wavelet-based signal processing, such as compression, takes advantage of the signal compaction achieved by the wavelet transform. Significant information is contained in a *few* sparsely located coefficients with the largest magnitudes. Coupled with this powerful property, very successful image compression methods make use of the multi-scale aspects of the transform; significant wavelet coefficients persist across wavelet scales. Jerome Shapiro's embedded zero-tree wavelet compression algorithm is a well known example [77], which takes advantage of the multi-scale and sparse aspects of the wavelet transform allowing for progressively better signal reconstruction as more significant coefficients are retained. There are a host of wonderful properties which make the wavelet transform a powerful signal processing tool.

The same success that compression algorithms have with largest magnitude coefficient selection have not been shared with classification problems [7, 9, 10]. Our relevance-wavelet model selects wavelet coefficients based on their GRLVQ-determined *importance* for the given classification which is a paradigm shift from more typical largest magnitude selection of coefficients. The significance of the relevance-wavelet model is that wavelet coefficients are selected linearly, where more typical applications select coefficients non-linearly (i.e., largest magnitudes). This paradigm shift results in very high classification accuracies with only a handful of wavelet coefficients.

7.3 Possibility for future work

We have taken the opportunity to present areas of possible future work throughout this thesis document. We suggested that one may be able to better determine how many prototype vectors to assign to each class in a GRLVQ(I) classifier by using information from a Self-Organizing Map (SOM) representation of the same data. The number of prototypes per class affects the classification which ultimately affects which input dimensions are deemed relevant for classification. As with many other feature ranking algorithms, determining how many features to retain is another open research topic for GRLVQ(I) learning. Closely related to feature selection was the topic of how best to evaluate those features. We suggested feature evaluation may be more appropriate with a different classifier, such as the K-means classifier, since the philosophy of a K-means classifier is more closely related to that of GRLVQ(I) than the Minimum Euclidean Distance (MED) classifier. In our investigation of wavelets as a feature space for GRLVQI processing, we found that the imaginary part of the DTCWT allowed GRLVQI to achieve better classification and feature extraction performance. Although we hypothesized that odd wavelet basis functions, in general, are better suited for subsequent GRLVQI processing for feature extraction and classification, the verification of this hypothesis is a topic for future research. Each of these topics are interesting, open topics, for continued research. We take this opportunity to present a few more interesting ideas which are larger in scope.

We previously discussed LVQ configuration issues with respect to the assignment of prototype vectors per class. In Section 3.5.1 we proposed using a clustering of the data with the Self-Organizing Map to guide this design consideration. Instead of trying to optimize the number of prototype vectors for each class ahead of time, one could start classifier training with a single prototype for each class and add them as they are needed. This idea is

motivated by Poirier and Ferrieux's Dynamic LVQ [78] and Bauer and Villmann's Growing Self-Organizing Map [79]. A Dynamic GRLVQI (DGRLVQI) is certainly a reasonable extension to the GLVQ family of classifiers (i.e., GRLVQI and GRLVQ). It is especially feasible given that GRLVQI descends a cost function which can be monitored throughout the training process. The idea is to train the classifier until the prototypes reach some stationary state and then add a prototype to the class which has the largest (average) contribution to the cost C of Eq. 2.5. The new prototype vector should be initialized in such a manner as to reduce the contribution of the given class to the total cost C. The advantage of this approach is that each class will have precisely the number of prototype vectors it needs to define its boundaries with its neighbors.

A successful classification of the data gives us an idea of how well the prototype vectors approximate the true decision boundary. However, it is impossible in its current high-dimensional form (in the case of the 194-dimensional Lunar Crater Volcanic Field data set evaluated in this thesis) to visualize the relationship between the converged prototype vectors and the boundary defined with neighboring classes. There are ways to project the data and the prototypes to a lower dimensional space so we can view the relationship between the data and the classification boundaries defined by the prototype vectors. One method which may be of practical use is a Sammon's mapping [80]. Sammon maps high-dimensional data to lower dimensions in an iterative fashion while preserving, on the lower dimensional mapping, the relationship of the data in its original high-dimensional form. A more principled alternative would map the converged prototype vectors from GRLVQ(I) (likely only those that have samples assigned to them) to the closest converged prototype [81] vector of a two-dimensional Self-Organizing Map [2] (SOM) representation of the data. The converged SOM preserves the topology from the original high-dimensional data to the lower two-dimensional SOM lattice [82–84]. More importantly, there is a direct relationship between the probability

distribution of the original data to that which is represented by the SOM (see [81, 85, 86]). The placement of the GRLVQ(I) prototype vectors on the SOM lattice would allow one to evaluate the quality of the classification boundary defined by GRLVQ(I) and may also assist in a better clustering of the data with the SOM. This interplay between the distribution of the data represented by the SOM and the class boundary definition by the GRLVQ(I) prototypes could provide valuable insights and is an open topic for continued research.

One subject that arose in our results was that of negative relevances when inverting the wavelet relevance factors. This begs the question: why not allow negative relevance factors? If we follow the ideas of linear spectral mixing proposed by Adams et al. [87–89], then the idea of negative relevance might be something to consider. In this view of spectral mixing analysis, negative fractions are allowed and are interpreted to mean that the model is non-physical, even if the model error is zero. In the case of negative relevance, we do not envision the same model interpretation. Our belief is that negative relevance would de-emphasize the contribution of the corresponding input dimensions' influence on the selection and updating of the prototypes for classification. The end result could be a better classification of the data, further reduction of retained features, or both. However, it is not clear how best to approach this problem at this time and is left as an area of continued research.

Although our wavelet-relevance model using the OSDWT produces superior classification and feature extraction results, it is possible a better wavelet model for hyperspectral images exists which will do a better job at preserving the important features while minimizing the effects of the discontinuities. One thought is to find the sparsest set of wavelet coefficients that, when reconstructed, matches that part of the signal we already know. In this model, we do not care what the reconstructed signal looks like in the wavelength bands that have corrupt data [90]. The advantages of such a model may not be realized until very difficult problems, even more difficult problems than what was presented in this thesis, are encountered.

Bibliography

[1] Barbara Hammer and Thomas Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15:1059–1068, 2002.

[2] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag Berlin Heidelberg, third edition, 2001.

[3] Michael J. Mendenhall. The airfield confidence map: Fusing hyperspectral and digital elevation data. Elec599 research project, Rice University, Apr 2004. Sponsored by Air Force Research Laboratories, Sensors Directorate, Wright-Patterson AFB, OH.

[4] Michael J. Mendenhall. Hyperspectral Field Analysis with specTIR Corp. Internal technical report, Air Force Research Laboratories, Sensors Directorate, Wright-Patterson AFB, OH., Sep 2004.

[5] Michael J. Mendenhall and Erzsébet Merényi. Relevance-based feature extraction from hyperspectral images. *IEEE Transactions on Neural Networks*, Mar 2006. Submitted.

[6] Michael J. Mendenhall and Erzsébet Merényi. On the Performance Comparison of two GRLVQ-type Classifiers for Hypersepctral Data. *IEEE Transactions on Geosciences and Remote Sensing*, 2006. In Preparation.

[7] Michael J. Mendenhall and Erzsébet Merényi. Generalized relevance learning vector quantization for classification-driven feature extraction from hyperspectral data. In *Proc. American Society for Photogrammetry and Remote Sensing, Reno, NV*, May 1–5, 2006.

[8] Michael J. Mendenhall and Erzsébet Merényi. Relevance-based feature extraction from hyperspectral images in the complex wavelet domain. *IEEE Mountain Workshop on*

Adaptive and Learning Systems, Mar 2006. Submitted.

[9] Tom Moon and Erzsébet Merényi. Classification of hyperspectral images using wavelet transforms and neural networks. In Andrew F. Laine, Michael A. Unser, and Mladen V. Wickerhauser, editors, *Proc. of the SPIE: Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 725–735, September 1995.

[10] Xudong Zhang, Nicolas H. Younan, and Charles G. O'Hara. Wavelet domain statistical hyperspectral soil texture classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):615–618, Mar 2005.

[11] J. A. Benediktsson, J. R. Sveinsson, and K. Arnason. Classification of very-high-dimensional data with geological applications. In *Proceedings of MAC Europe 91*, pages 13–18, October 1994.

[12] Karen L. Oehler and Robert M. Gray. Combining image compression and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):461–473, May 1995.

[13] Thomas Villmann, Erzsébet Merényi, and Barbara Hammer. Neural maps in remote sensing image analysis. *Neural Networks*, 16:389–403, 2003.

[14] Koby Crammer, Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Margin analysis of the LVQ algorithm. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 462–469, Cambridge MA, 2002. MIT Press.

[15] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, second edition, 2001.

[16] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference*, pages 423–429, Cambridge, MA, 1996. MIT Press.

[17] Atsushi Sato and Keiji Yamada. A formulation of learning vector quantization using a new misclassification measure. In *Proceedings of the Fourteenth International Conference on Pattern Recognition*, volume 1, pages 322–325, Aug 1998.

[18] Biing-Hwang Juang and Shigeru Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40(12):3043–3054, Dec 1992.

[19] D. DeSieno. Adding a conscience to competitive learning. In *Proceedings of the IEEE International Conference on Neural Networks* **I**, pages 117–124, New York, Jul 1988.

[20] Stanley C. Ahalt, Ashok K. Krishnamurthy, Prakoon Chen, and Douglas E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3:277–290, 1990.

[21] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[22] Erzsébet Merényi. Precision mining of high-dimensional patterns with self-organizing maps: Interpretation of hyperspectral images. In Peter Sincak and Jan Vascak, editors, *Quo Vadis Computational Intelligence: New Trends and Approaches in Computational Intelligence. Studies in Fuzziness and Soft Computing.*, volume 54. Physica-Verlag, 2000. Available: http://www.ece.rice.edu/~erzsebet/publications.html.

[23] James B. Campbell. *Intro. to Remote Sensing.* Guilford Press, 2 edition, 1996.

[24] E.S. Howell, E. Merényi, and L. A. Lebofsky. Classificaiton of asteroid spectra using a neural network. *Journal of Geophysical Research*, 99(E5):10,847–10,865, May 1994.

[25] Graham R. Hunt and John W. Salisbury. Visible and near-infrared spectra of minerals and rocks: I. Silicate Minerals. *Modern Geology*, 1:283–300, 1970.

[26] Graham R. Hunt and John W. Salisbury. Visible and near-infrared spectra of minerals and rocks: II. Carbonates. *Modern Geology*, 2:23–30, 1971.

[27] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: III. Oxides and Hydroxides. *Modern Geology*, 2:195–205, 1971.

[28] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: IV. Sulphides and Sulphates. *Modern Geology*, 3:1–14, 1971.

[29] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: V. Halides, Phosphates, Arsenates, Vanadates, and Borates. *Modern Geology*, 3:121–132, 1972.

[30] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: VI. Additional Silicates. *Modern Geology*, 4:85–106, 1973.

[31] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: VII. Acidic Igneous Rocks. *Modern Geology*, 4:217–224, 1973.

[32] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: VIII. Acidic Igneous Rocks. *Modern Geology*, 4:237–244,

1973.

[33] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: IX. Basic and Ultrabasic Igneous Rocks. *Modern Geology*, 5:15–22, 1974.

[34] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: X. Stony Meteorites. *Modern Geology*, 5:115–126, 1975.

[35] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: XI. Sedimentary Rocks. *Modern Geology*, 5:211–217, 1976.

[36] Graham R. Hunt, John W. Salisbury, and Charles J. Lenhoff. Visible and near-infrared spectra of minerals and rocks: XII. Metamorphic Rocks. *Modern Geology*, 5:219–228, 1976.

[37] Ted Roush and Robert B. Singer. Gaussian Analsyis of Temperate Effects on the Reflectance Spectra of Mafic Minerals in the 1-μm Region. *Journal of Geophysical Research*, 91(B10):10,301–10,308, 1986.

[38] Howard P. Ross, Joel E. M. Adler, and Graham R. Hunt. A Statistical Analysis of the Reflectance of Igneous Rocks from 0.2 to 2.65 Microns. *ICARUS*, 11:46–54, 1969.

[39] R. O. Green. Summaries of the 6th Annual JPL Airborne Geoscience Workshop, 1. AVIRIS Workshop. Pasadena, CA, Mar 4–6 1996.

[40] Erzsébet Merényi, R.B. Singer, and J.S. Miller. Mapping of spectral variations on the surface of mars from high spectral resolution telescopic images. *ICARUS*, 124:280–295, 1996.

[41] E. Merényi, J.V. Taranik, T.B. Minor, and W.H. Farrand. Quantitative comparison of neural network and conventional classifiers for hyperspectral imagery. In R.O. Green, editor, *Summaries of the 6th Annual JPL Airborne Geoscience Workshop, 1. AVIRIS Workshop*, volume 1, Pasadena, CA, Mar 4–6 1996.

[42] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer, Upper Saddle River, New Jersey 07458, 2001.

[43] Simon Haykin. *Neural Network – A Comprehensive Foundation.* Prentice Hall, Upper Saddle River, New Jersey 07458, second edition, 1999.

[44] Jari A.Kangas, Teuvo K.Kohonen, and Jorma T. Laaksonen. Variants Self-Organizing Maps. *IEEE Transactions on Neural Networks*, 1(1):93–99, Mar 1990.

[45] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition.* Academic Press, first edition, 1999.

[46] P.L. Bartlett and S. Mendelson. Radamacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning and Research*, 3:463–482, 2002.

[47] Barbara Hammer, Marc Strickert, and Thomas Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, 21(2):109–120, Apr 2005.

[48] Mu-Chun Su, Ta-Kang Liu, and Hsiao-Te Chang. An efficient initialization scheme for the self-organizing feature map algorithm. In *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pages 1906–1910, Jul 1999.

[49] Atsushi Sato. An analysis of initial state dependence in generalized LVQ. In *Ninth International Conference on Artificial Neural Networks.*, volume 2, pages 928–933, Sep 1999.

[50] Janez Demšar. Statistical Comparisons of Classifier over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 2006.

[51] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[52] Shaomin Wu and Peter Flach. A scored AUC metric for classifier evaluation and selection. *International Conference on Machine Learning*, 2005.

[53] Robert C. Holte and Chris Drummond. Cost-sensitive classifier evaluation. In *UBDM '05: Proceedings of the 1st international workshop on Utility-based data mining*, pages 3–9, New York, NY, USA, 2005. ACM Press.

[54] Sophia Daskalaki, Ioannis Kapanas, and Nikolas Avouris. Evaluation of classifiers for an uneven class distribution problem. *Applied Artificial Intellegence*, pages 1–24, 2006. Draft manuscript accepted for publication.

[55] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[56] Pedro Domingos and Foster Provost. Well-trained PETs: Improving probability estimation trees, 2000. CDER Working Paper 00-04-IS, Stern School of Business, New York University, NY, NY 10012.

[57] D. J. Hand and R. J. Till. A simple generalization of the area uder the ROC curve to multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[58] N. M. Adams and D. J. Hand. Improving the practice of classifier performance assessment. *Neural Computation*, 12:305–311, 2000.

[59] Thorsten Bojer, Barbara Hammer, Daniel Schunk, and Katharina Tluk von Toschanowitz. Relevance Determination in Learning Vector Quantization. In *Proceedings of European Symposium on Artificial Neural Networks (ESANN01)*, pages 271–276, Bruges, Belgium, Apr 2001. D facto publications.

[60] Thomas M. Lillesand and Ralph W. Kiefer. *Remote Sensing and Image Interpretation.* John Wiley & Sons, Inc., 4 edition, 2000.

[61] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, XX(1):37–46, 1960.

[62] Yvonne M. M. Bishop, Stephen E. Fienberg, and Paul W. Holland. *Discrete Multivariate Analysis: Theory and Practice.* The MIT Press, 1975. with the collaboration of Richard J. Light and Frederick Mosteller.

[63] Steven L. Salzberg. Statistical Comparisons of Classifier over Multiple Data Sets. *Data Mining and Knowledge Discovery*, 1:317–328, 1997.

[64] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, Dec 1945.

[65] Frank Wilcoxon. Probability Tables for Individual Comparisons by Ranking Methods. *Biometrics*, 3(3):119–122, Sep 1947.

[66] Robert L. McCornack. Extended Tables of the Wilcoxon matched Pair Signed Rank Statistic. *Journal of the American Statistical Association*, 60(311):864–871, Sep 1965.

[67] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-SNF Regional Conference Series on Applied Mathematics.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[68] C. Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo. *Introduction To Wavelets and Wavelet Transforms: A Primer.* Prentice-Hall, Englewood Cliffs, NJ, 1997.

[69] H.L. Resnikiff and R.O. Wells. *Wavelet Analysis.* Springer-Verlag, New York, 1998.

[70] M. Vetterli and C. Herley. Wavelets and filterbanks: Theory and design. *IEEE Transactions on Acoustic and Speech Signal Processing*, 40(9):2207–2232, 1992.

[71] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[72] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:710–732, Jul 1992.

[73] S. Mallat and W. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, 1992.

[74] P.P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[75] Ivan W. Selesnick, Richard G. Baraniuk, and Nick G. Kingsbury. The Dual-Tree Complex Wavelet Transform. *IEEE Signal Processing Magazine*, pages 123–151, Nov 2005.

[76] Nick G. Kingsbury. A dual-tree complex wavelet transform with improved orthogonality and symmetry properties. In *Proceedings of the IEEE International Conference on Image Processing (ICIP), 2000*, volume 2, pages 375–378, September 2000.

[77] Jerome M. Shapiro. Embedded image encoding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993.

[78] F. Poirier and A. Ferrieux. DVQ: Dynamic Vector Quantization - An Incremental LVQ. In *International Conference on Artificial Neural Networks '91*, pages 1333–1336, 1991.

[79] Hans-Ulrich Bauer and Thomas Villmann. Growing a hypercubical output space in a self-organizing feature map. *IEEE Transactions on Neural Networks*, 8(2):218–226, Mar 1997.

[80] John W. Sammon Jr. A Nonlinear mapping for data Structure Analysis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969.

[81] H. Ritter and K. Schulten. On the Stationary State of Kohonen's Self-Organizing Sensory Mapping. *Biological Cybernetics*, 54:99–106, 1986.

[82] Hans-Ulrich Bauer and Klaus R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4):570–579, 2001.

[83] Thomas Martinetz and Klaus Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[84] Thomas Villmann, Ralf Der, Michael Herrmann, and Thomas M. Martinetz. Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, Mar 1997.

[85] Helge Ritter. Asymptotic level density for a class of vector quantization process. *IEEE Transactions on Neural Networks*, 2(1):173–175, Jan 1991.

[86] Dominik R. Dersch and Paul Tavan. Asymptotic Level Density in Topological Feature Maps. *IEEE Transactions on Neural Networks*, 6(1):230–236, Jan 1995.

[87] J. B. Adams, M. O. Smith, and P. E. Johnson. Spectral mixture modeling - A new analysis of rock and soil types at the Viking Lander 1 site. *Journal of Geophysical Research*, 91:8098–8112, Jul 1986.

[88] Robert B. Singer and Thomas B. McCord. Mars: Large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance. In *Proceedings of the 10th Lunar Planetary Science Conference*, pages 1835–1848, 1979.

[89] John B. Adams, Milton O. Smith, and Alan R. Gillespie. Imaging Spectroscopy: Interpretation Based on Spectral Mixture Analysis. In Carle M. Pieters and Peter A. Englert, editors, *Remote Geochemical Analysis: Elemental and Mineralogical Composition*, pages 145–166. Cambridge University Press, 1993.

[90] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk. Distributed compressed sensing. *IEEE Transactions on Information Theory*, 2005. Submitted.